



**Treball de Fi de Grau**

**GRAU D'ENGINYERIA INFORMÀTICA**

**Facultat de Matemàtiques  
Universitat de Barcelona**

---

# **Web para estudio dermatológico**

---

**Víctor García Delgado**

**Director:** Simone Balocco  
**Realizado en:** Departament de  
Matemàtica i Informàtica. UB  
Barcelona, 1 de febrero de 2019



# Índice

<b>Resumen (castellano).....</b>	<b>5</b>
<b>Resumen (catalán).....</b>	<b>6</b>
<b>Resumen (inglés).....</b>	<b>7</b>
 <b>1. Introducción.....</b>	 <b>8</b>
1.1 Contexto.....	9
1.2 Motivación.....	10
 <b>2. Planificación del TFG.....</b>	 <b>11</b>
2.1 Oferta o idea inicial.....	11
2.2 Primera propuesta.....	12
2.3 Propuesta final.....	14
2.4 Comparativa entre planificaciones.....	16
 <b>3. Análisis de requerimientos.....</b>	 <b>18</b>
3.1 Primer análisis.....	18
3.2 Requisitos funcionales.....	18
3.3 Requisitos no funcionales.....	23
 <b>4. Diseño del modelo de datos.....</b>	 <b>25</b>
4.1 Modelo Entidad/Relación.....	25
 <b>5. Tecnología.....</b>	 <b>26</b>
5.1 Servidor.....	26
5.2 Servidor Web - Apache.....	27
5.3 Lenguaje de programación web - PHP.....	28
5.4 Gestor de dependencias - Composer.....	29
5.5 Framework PHP - Laravel.....	30
5.6 Framework HTML & CSS – Bootstrap.....	31
5.7 Control de versiones – Git/Github.....	32
5.8 Editor de texto/IDE – Visual Studio Code.....	33
5.9 Administración y gestión – Trello.....	35
 <b>6. El framework Laravel.....</b>	 <b>36</b>
6.1 Componentes de Laravel.....	36
6.2 Sistema de plantillas Blade.....	37
6.3 Eloquent ORM.....	39
6.4 Routing y middlewares.....	41

<b>7. Estructura de la plataforma web.....</b>	<b>43</b>
7.1 Web corporativa.....	43
7.2 Área privada.....	44
<b>8. Desarrollando la aplicación.....</b>	<b>46</b>
8.1 Comparación entre servidores.....	46
8.1.1 Pruebas con los servidores.....	47
8.1.2 Tabla de servidores.....	49
8.2 Ejecución de un script Python en Laravel.....	49
8.3 Test Driven Development.....	50
<b>9. Posibles ampliaciones.....</b>	<b>51</b>
<b>10. Conclusiones.....</b>	<b>53</b>
<b>13. Bibliografía.....</b>	<b>54</b>
<b>14. Anexo.....</b>	<b>57</b>
14.1 Configuración del entorno.....	57
14.1.1 Servidor.....	57
14.1.2 Servidor HTTP Apache.....	60
14.1.3 PHP.....	63
14.1.4 MySQL.....	63
14.1.5 Composer.....	64
14.1.6 Git.....	65
14.1.7 PHPMyAdmin.....	65
14.1.8 Laravel.....	67
14.2 Casos de uso.....	71
14.2.1 Usuario invitado.....	71
14.2.2 Usuario por defecto.....	73
14.2.3 Doctor.....	77
14.2.4 Administrador.....	82
14.3 Manual de usuario.....	88
14.3.1 Web corporativa.....	88
14.3.2 Área privada - usuario.....	89
14.3.3 Área privada - doctor.....	97
14.3.4 Área privada - admin.....	108
14.4 Muestra de consulta.....	122

## Resumen: Castellano

A lo largo del presente documento se detalla el desarrollo del proyecto llamado “WDM: Web para estudio dermatológico”, basado en una plataforma web para el análisis dermatológico relacionada con la oferta de Trabajo Final de Grado<sup>[1]</sup> de la Universidad de Barcelona (UB)<sup>[2]</sup>, en concreto, con la Facultad de Matemáticas e Informática<sup>[3]</sup>. Este proyecto se realiza también en colaboración con el Hospital Italiano de Buenos Aires<sup>[4]</sup>, Argentina.

La finalidad de la web es el análisis y prevención del cáncer de piel que el usuario de la plataforma pueda tener. Del mismo modo, se ofrece la posibilidad de tener un histórico de consultas para poder ver la evolución de las mismas a lo largo del tiempo. Estas consultas pueden ser revisadas por los doctores que usan la plataforma, de modo que pueden introducir comentarios relacionados con la consulta que ha realizado el usuario.

Se ha intentado diseñar una plataforma flexible, por lo tanto se ha seguido una metodología enfocada a poder introducir posibles mejoras y/o evoluciones para poder perfeccionar la experiencia del usuario, así como futuros requerimientos el hospital pueda solicitar a otros compañeros de la universidad que quieran desarrollar su proyecto final sobre el que se describe a continuación.

A continuación se puede encontrar el detalle del análisis de requerimientos, diseño, planificación, así como diversos puntos relacionados con la plataforma, tales como las tecnologías usadas y la arquitectura o estructura de la web.

## Resumen: Català

Al llarg del present document es detalla el desenvolupament del projecte anomenat “WDM: Web per a estudi dermatològic”, basat en una plataforma web per a l'anàlisi dermatològica relacionada amb l'oferta de Treball Final de Grau<sup>[1]</sup> de la Universitat de Barcelona (UB)<sup>[2]</sup>, en concret, amb la Facultat de Matemàtiques i Informàtica<sup>[3]</sup>. Aquest projecte es realitza també en col·laboració amb l'Hospital Italià de Buenos Aires<sup>[4]</sup>, l'Argentina.

La finalitat de la web és l'anàlisi i prevenció del càncer de pell que l'usuari de la plataforma pugui tenir. De la mateixa manera, s'ofereix la possibilitat de tenir un històric de consultes per a poder veure l'evolució de les mateixes al llarg del temps. Aquestes consultes poden ser revisades pels doctors que usen la plataforma, de manera que poden introduir comentaris relacionats amb la consulta que ha realitzat l'usuari.

S'ha intentat dissenyar una plataforma flexible, per tant s'ha seguit una metodologia enfocada a poder introduir possibles millores i/o evolucions per a poder perfeccionar l'experiència de l'usuari, així com futurs requeriments l'hospital pugui sol·licitar a altres companys de la universitat que vulguin desenvolupar el seu projecte final sobre el qual es descriu a continuació.

A continuació es pot trobar el detall de l'anàlisi de requeriments, disseny, planificació, així com diversos punts relacionats amb la plataforma, tals com les tecnologies usades i l'arquitectura o estructura de la web.

## Resumen: English

This document detail the development of the project "WDM: dermatology website", based on a web platform for the dermatological analysis, as described in the offer of Final Degree Project<sup>[1]</sup> of the University of Barcelona (UB)<sup>[2]</sup>, specifically, within the Faculty of Mathematics and Informatics<sup>[3]</sup>. This project is also carried out in collaboration with the Hospital Italiano de Buenos Aires<sup>[4]</sup>, Argentina.

The purpose of the website is the analysis and early detection of skin cancers that the user of the platform may have. Similarly, it offers the possibility of having a history of queries to see the evolution of the skin lesion over time. These queries can be reviewed by the doctors who use the platform, so they can interact with the user and enter comments related to the query that the user has made.

To carry out the development of the platform, the platform was developed as flexible as possible, hence it was followed a methodology focused on introducing possible improvements and/or evolutions to improve the user experience, as well as future requirements the hospital may request. Finally the web allow to include future development of the web carried out by other colleagues of the university who want to develop their final project using this framework.

Below you can find the detail of the requirements analysis, design, planning, as well as various points related to the platform, such as the technologies used and the architecture or structure of the web.

# 1. Introducción

A lo largo del tiempo la esperanza de vida del ser humano ha ido en aumento. En una primera etapa, esta mejora venía relacionada con la adquisición o la posibilidad de las personas de poder cambiar ciertos hábitos o situaciones. Pero en la etapa actual ha sido esencial la evolución de la ciencia, que ha aportado a la sociedad muchos avances e innovaciones. Estos avances han permitido que las personas pudieran tanto curarse de enfermedades que en otro momento no hubiese sido posible o bien, incluso, evitar directamente la enfermedad o problema.

Precisamente, el hecho de poder evitar contraer una enfermedad o tener cualquier tipo de problema es el más importante. En primer lugar, porque aunque la ciencia ha avanzado de una manera considerable, nunca se llegará a alcanzar la perfección (siempre refiriéndose a que será completamente efectiva), por lo que hay muchas enfermedades de las cuales aún no se sabe lo suficiente. Esto deriva en que las causas y/o los remedios del problema son desconocidos, y esto ocurre en casos que son, desgraciadamente, muy comunes (distintos tipos de cáncer, alzheimer...). Y en segundo lugar, porque es el “remedio” más importante. Aunque suena trivial, no es necesario curarse de una enfermedad que no se tiene. Y en esos pasos son en los que avanza la ciencia.

Aún así, hay un punto intermedio entre el prevenir contraer una enfermedad y esperar realmente a una solución efectiva a la perfección: el diagnóstico precoz. El hecho de poder detectar un problema en su fase inicial (incluso sin síntomas) hace que las posibles vías de actuación sean mayores y mucho más efectivas. Para ello la mayoría de las veces se precisa de algún tipo de chequeo concreto que, en caso de detectar algún problema precoz, aporta muchos puntos positivos al paciente<sup>[5]</sup>. En este sentido hay estudios<sup>[6]</sup> que certifican la importancia de este punto.

El principal problema de poder obtener un diagnóstico precoz de cualquier tipo de enfermedad, en muchos casos, es el hecho de tener la posibilidad de que un doctor o un experto en concreto realice algún tipo de exploración al interesado. Un hecho que, teniendo en cuenta la actual masificación de sistemas sanitarios como el español provocan que muchas personas no consigan una revisión a tiempo.

Ante esta situación toman mucha importancia las nuevas tecnologías que permiten realizar, de una manera más independiente, muchos diagnósticos y comprobaciones. En este último punto ese es el que se centra el proyecto.



## 1.1 Contexto

La sociedad actual está altamente caracterizada, dentro de los avances científicos en general, por la gran importancia de la informática y nuevas tecnologías. Esta importancia se refleja en muchos campos (la robótica en las fábricas por ejemplo), pero en especial, en el de la medicina.

En la medicina, la informática ha sido recibida con los brazos abiertos. Muchos de los instrumentales médicos han heredado los distintos avances que se han ido consiguiendo para mejorar el bienestar de todas las personas. Esto se traduce en máquinas que pueden ofrecer más detalles a los doctores así como unos diagnósticos más precisos y acertados. Es bastante frecuente que muchos procesos que anteriormente se hacían de un modo “manual”, sea actualmente realizado de manera automática (con y sin la intervención de un doctor).

Avanzando un poco más se ha llegado al punto de poder aportar a estas máquinas de una Inteligencia Artificial que haga que por sí mismas puedan tanto diagnosticar como aprender de todos sus diagnósticos<sup>[7]</sup>. Cabe la posibilidad de que en un futuro sean solamente estas máquinas quienes se encarguen de realizar este tipo de procesos (aquí ya entraríamos en cuestiones éticas y morales), pero lo que sí que es cierto es que el uso de estas máquinas con IA como una extensión del doctor<sup>[8]</sup> (y no una sustitución), hace que el trabajo sea más efectivo al poder realizarse muchos más diagnósticos y poder abarcar a una población mayor.

Y es que el mundo tecnológico actual tiene muchos aspectos negativos también, pero hay alguno de estos aspectos que se pueden utilizar para beneficio propio. Esta referencia es al denominado Big Data. Hay muchísimos datos que en ocasiones son utilizados para poder ofrecer a una persona un anuncio más “relacionado” con su conducta o recomendar una película que “seguro” que le gusta en función de lo que ha podido ir viendo, aportando una visión a la persona de “sentirse vigilada”, o de que “saben todo sobre ella”. Pero, como se ha dicho, puede utilizarse toda esa información para fines médicos.

Llegados a un punto en el que las máquinas pueden ayudar a procesar todo ese tipo de datos y obtener patrones o conclusiones de una manera eficiente<sup>[9]</sup>, el no utilizar ese recurso no es más que apartar la vista de una gran ayuda. Es por este motivo que cualquier tipo de proceso o herramienta automática que se le pueda ofrecer a la población para el diagnóstico de algo en concreto, con una eficiencia aceptable y sin la presencia de un doctor, puede ayudar a que todas las personas tengan la posibilidad de anticiparse a cualquier tipo de enfermedad y poder actuar de una mejor manera.

## 1.2 Motivación

En la realización de este Trabajo Final de Grado ha influido de manera notoria la idea<sup>[1]</sup> aportada por el profesor Simone Balocco, quien de un modo muy claro pudo exponer la idea para que el alumno la pudiese analizar.

Tras un análisis, surgieron una serie de motivaciones para la realización de esta idea:

- En primer lugar, viendo todo el antecedente explicado en la primera parte de la introducción, resultaba algo positivo poder desarrollar una plataforma web con el objetivo de que los usuarios pudiesen realizar un análisis de manera independiente que les pudiese facilitar el saber el estado de algún problema que pudieran tener.
- En segundo lugar, el hecho de poder trabajar con un cliente real, en este caso el Hospital Italiano de Buenos Aires, acerca este proyecto a un entorno más real. Aunque sea de un modo académicamente controlado, permite al alumno ponerse en situaciones en las que se verá en el entorno laboral.
- Por otro lado, a nivel personal, permite seguir la línea de investigación que quería seguir el alumno: el ámbito web. En este punto puede aportar alguna experiencia relacionada con algunas asignaturas de la carrera así como experiencias laborales previas. En este sentido quizás las experiencias no se asemejan completamente, pero suponen un punto de partida interesante. Además, estas experiencias datan de algunos años atrás, por lo que permitirá ver la evolución sufrida en el sector.
- Por último, cumplía también otro deseo del alumno como es el aprovechar este trabajo para aprender alguna tecnología no usada anteriormente. En este caso, el framework PHP Laravel.

## 2. Planificación del TFG

El TFG desarrollado y explicado en el siguiente documento, se ha realizado con la ayuda de una serie de reuniones periódicas con José Massa, profesor de la Universidad UNICEN<sup>[10]</sup>. José es responsable del proyecto dermatológico del Hospital Italiano de Buenos Aires, y ha proporcionado su feedback a lo largo del proyecto. Por esta razón se han tomado una serie de decisiones y cambios de rumbo en función de las necesidades del proyecto.

### 2.1 Oferta o idea inicial

La idea propuesta como Trabajo Final de Grado (TFG), en su etapa inicial, está basada en una herramienta o aplicación web para el Hospital Italiano de Buenos Aires. La finalidad de esta herramienta es el análisis automático, mediante un algoritmo, de un lunar en la piel del usuario que haga uso de la plataforma. Mediante la subida de una imagen clara del lunar a analizar, el algoritmo determina su estado.

Para que un usuario pudiese utilizar la aplicación debería rellenar un formulario con sus datos personales (tales como nombre, edad...etc) y con algunos datos relacionados con la foto (como la zona del cuerpo en la que se ha realizado). Todos estos datos ayudarán en el algoritmo. Una vez el usuario ha rellenado dichos datos y ha seleccionado la foto a utilizar, el algoritmo se pondría en marcha aportando un resultado al finalizar. Este resultado sería siguiendo la estructura de semáforo:

- **Verde:** El algoritmo determina que no hay riesgo de cáncer en el lunar de la imagen.
- **Amarillo:** El algoritmo detecta un posible riesgo, pero con una probabilidad baja, por lo cual recomienda una visita médica para su revisión.
- **Rojo:** El algoritmo detecta que hay un riesgo de cáncer elevado en el lunar de la imagen y remite a una visita médica inmediata.

Es importante destacar que el algoritmo que realiza el análisis del lunar no es parte de la investigación de este TFG. El objetivo es realizar una plataforma web que permita la recogida de imágenes de pacientes, donde se pueda ejecutar el código que realiza el análisis automático de la imagen. El resultado de dicho análisis será visualizado por el paciente y almacenado para futuras consultas. El algoritmo que realiza el análisis automático será objeto de futuras investigaciones del grupo y futuros TFG.

Debido que en la aplicación se usarán datos personales es necesario otorgar a la plataforma las medidas necesarias para que estos datos puedan estar seguros.

Es posible ver la oferta concreta en la referencia<sup>[1]</sup>.

## 2.2 Primera propuesta

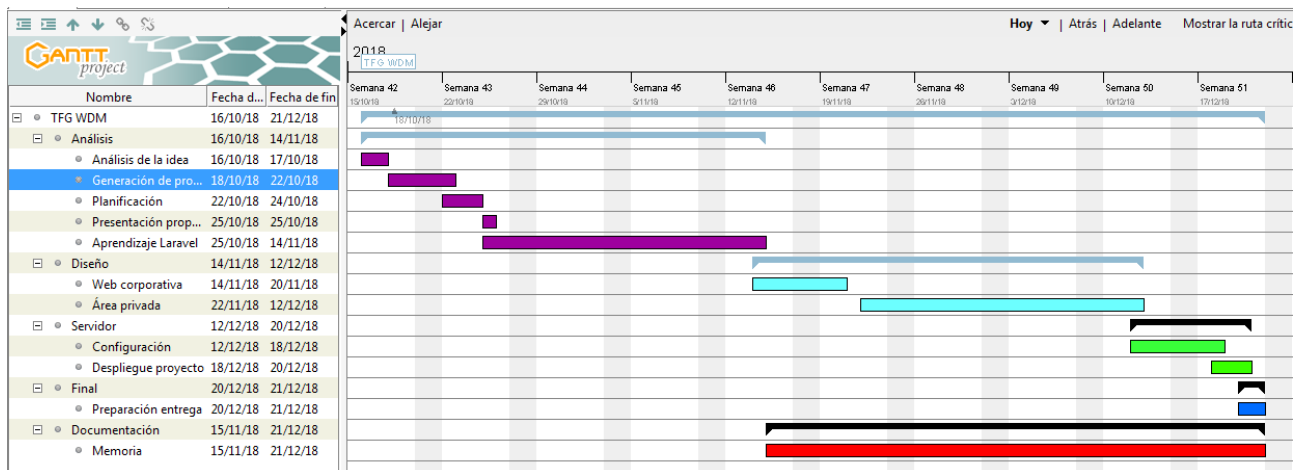
Partiendo de la de la idea planteada como propuesta de TFG, se ha realizado una consideración inicial para determinar una posible estructura para la aplicación. La funcionalidad de la herramienta sigue exactamente las características necesarias, pero se añade algún detalle con la finalidad de extender la idea.

Entrando en detalle, se describe la herramienta en dos partes diferenciadas:

- Por un lado se tendría el punto de entrada a la aplicación, que consistiría en una página web de tipo corporativa. En dicha página se puede introducir información acerca del Hospital Italiano y/o del proyecto dermatológico en sí. Además es el lugar adecuado para que los usuarios puedan encontrar distintas informaciones de contacto con el departamento dermatológico. De este modo el punto de entrada en la aplicación no sería directamente el aplicativo para subir la imagen a analizar.
- Por otro lado se tendría el formulario de subida de la imagen a revisar. En esta sección se requeriría al usuario los datos personales necesarios así como datos informativos sobre la imagen subida, además de dicha imagen.

Si se establece esta estructura es posible aprovechar las características de los sitios web y aportar información sobre el proyecto dermatológico y otras posibles informaciones de cara al usuario. De este modo no es visible el aplicativo web nada más entrar, aportando un toque de dinamismo y de movimiento.

A continuación se presenta del diagrama de Gantt con la planificación:



*Ilustración 1: Diagrama de Gantt de la planificación inicial*

La ilustración 1 muestra el diagrama de Gantt con la planificación ideal del proyecto. Cabe destacar que la planificación se ha realizado teniendo en cuenta que el primer día de proyecto el alumno lo cuenta desde el día que aceptó realizar este proyecto (16 de octubre de 2018), hasta la fecha acordada con el profesor Simone Balocco para realizar una entrega de revisión (21 de diciembre de 2018).

A continuación las etapas definidas de 16/10/2018 a 21/12/2018:

#### **Análisis (16/10 a 14/11)**

- Análisis de la idea: 16/10 a 17/10
- Generación de la propuesta: 18/10 a 22/10
- Planificación: 22/10 a 24/10
- Presentación de la propuesta: 25/10 a 25/10
- Aprendizaje Laravel: 25/10 a 14/11

#### **Servidor (12/12 a 20/12)**

- Configuración: 12/12 a 18/12
- Despliegue proyecto: 18/12 a 20/12

#### **Documentación (15/11 a 21/12)**

- Memoria: 15/11 a 21/12

#### **Diseño (14/11 a 12/12)**

- Web corporativa: 14/11 a 20/11
- Área privada: 22/11 a 12/12

#### **Final (20/12 a 21/12)**

- Preparación entrega: 20/12 a 21/12

## 2.3 Propuesta final

Después de realizar un análisis más profundo sobre las posibilidades que tiene la idea planteada y, después de revisar la propuesta inicial, se ha llegado a la conclusión de que se podría introducir algunos pequeños cambios más de cara a aprovechar la herramienta correctamente.

Partiendo de la base de que la aplicación tiene como objetivo que, de una manera automática, se pueda analizar algún lunar en la piel para poder actuar lo más rápido posible en caso de que se requiera, sería interesante introducir una funcionalidad de tipo “historial”. Si esto se realiza es posible tener un control más exhaustivo de la evolución de un mismo lunar, aspecto muy importante a la hora de poder anticiparse a cualquier problema. Por otro lado, introduciendo esta característica se aprovecharía la base de datos (BBDD) que habría detrás de la aplicación.

La estructura de la herramienta quedaría de la siguiente manera:

- Se mantiene el punto de entrada a la aplicación como una web corporativa a modo de presentación y de informativo sobre el proyecto dermatológico.
- Se propone introducir un área privada en la plataforma. El acceso a dicha área sería mediante un usuario y contraseña, con dos roles diferenciados de usuarios: usuario público y usuario doctor.

La finalidad de introducir el área privada es, en primer lugar, aportar un grado más de seguridad en la herramienta, ya que el aplicativo de análisis de imagen no estaría abierto a todo tipo de usuarios, si no solamente a los registrados. Con este cambio, posibles ataques con robots o bots automáticos que pudiesen atacar formularios de forma masiva quedarían en cierto modo salvados, ya que no tendrían acceso.

Una vez logeado, el usuario público podría ver un listado con todas sus consultas agrupadas por la zona de la imagen de la consulta. Esto quiere decir, por ejemplo, que todas las consultas de un lunar de la mano derecha quedarían agrupadas bajo una misma entrada. Al seleccionar dicha entrada se accedería al listado completo de las consultas relacionadas con el mismo lunar de la mano derecha. De esta manera es posible ver la evolución de una manera más sencilla.

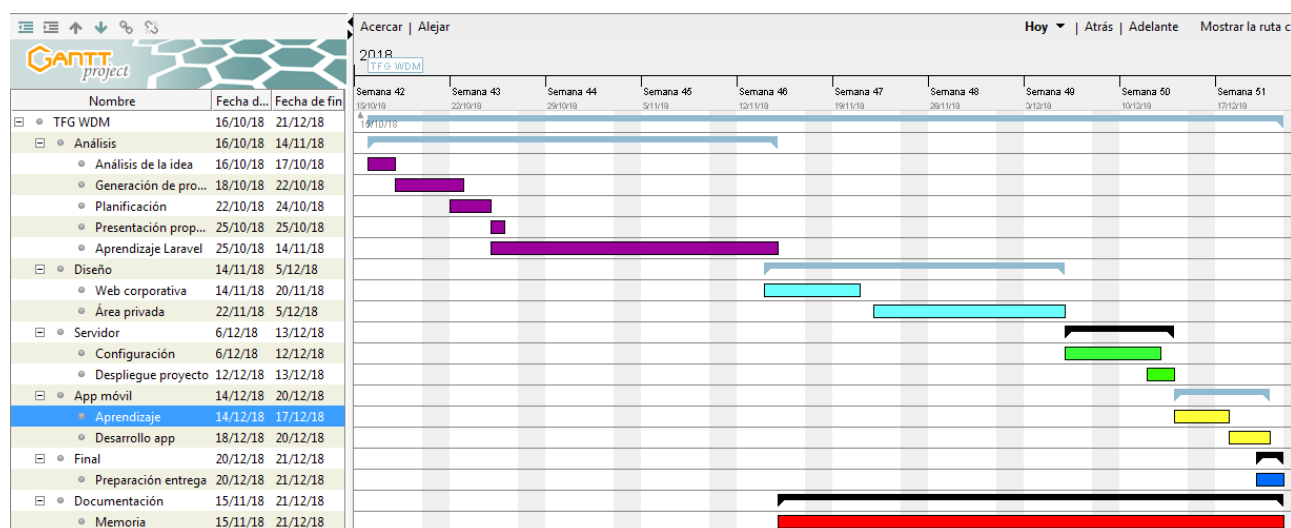
Además, como es normal, el usuario podrá realizar una nueva consulta (ya sea de una zona nueva o de una ya realizada). Aprovechando que el usuario tendrá un perfil con sus datos, ya no será necesario que los introduzca. Es suficiente que la consulta la haga indicando si está relacionada con una consulta antigua (seleccionando en un listado qué consulta sería), o bien sería una consulta nueva (indicando, entonces, la zona). Esto aportaría a la herramienta un plus de accesibilidad y/o funcionamiento, ya que se evita solicitar cada vez los datos personales al usuario.

Por otro lado, si el usuario logeado es doctor, además de disponer de un módulo CRUD (creación, edición y eliminación de usuarios), podrá ver un listado de las consultas realizadas por los usuarios, con la posibilidad de realizar en cada una de ellas una pequeña reseña en un campo de

texto. De este modo se introduciría en la herramienta la interacción personal de los doctores dermatólogos además del análisis automático del algoritmo.

Todo el análisis de propuesta se ha realizado en una etapa del proyecto donde el requerimiento era en todo momento web. Tras una reunión en una etapa avanzada (23 de noviembre de 2018), el cliente determina que el requerimiento sería enfocado en una APP móvil, y no en una plataforma web. Por este motivo y, tras el análisis del problema con el tutor, se ofrece la posibilidad de seguir con el desarrollo de la plataforma web para aprovechar la investigación que hasta la fecha por el alumno, y éste, a su vez, ofrece la solución de generar una APP (inicialmente en Android) usando el componente WebView<sup>[12]</sup>, que permite crear la APP con el menor impacto posible a nivel de código y tiempo.

A continuación se presenta el diagrama de Gantt con la planificación final (con los cambios de objetivos introducidos en las reuniones periódicas):



*Ilustración 2: Diagrama de Gantt de la planificación final*

La ilustración 2 muestra el diagrama de Gantt con la planificación ideal del proyecto. Cabe destacar que la planificación se ha realizado teniendo en cuenta que el primer día de proyecto el alumno lo cuenta desde el día que aceptó realizar este proyecto (16 de octubre de 2018), hasta la fecha acordada con el profesor Simone Balocco para realizar una entrega de revisión (21 de diciembre de 2018). Además, se han introducido los cambios de objetivos planteados en las reuniones periódicas con respecto de la planificación inicial.

A continuación las etapas definidas de 16/10/2018 a 21/12/2018:

#### **Análisis (16/10 a 14/11)**

- Análisis de la idea: 16/10 a 17/10
- Generación de la propuesta: 18/10 a 22/10
- Planificación: 22/10 a 24/10

#### **Diseño (14/11 a 5/12)**

- Web corporativa: 14/11 a 20/11
- Área privada: 22/11 a 5/12

- Presentación de la propuesta: 25/10 a 25/10

- Aprendizaje Laravel: 25/10 a 14/11

#### **Servidor (6/12 a 13/12)**

- Configuración: 6/12 a 12/12
- Despliegue proyecto: 12/12 a 13/12

#### **Final (20/12 a 21/12)**

- Preparación entrega: 20/12 a 21/12

#### **App móvil (14/12 a 20/12)**

- Configuración: 14/12 a 17/12
- Despliegue proyecto: 17/12 a 20/12

#### **Documentación (15/11 a 21/12)**

- Memoria: 15/11 a 21/12

## 2.4 Comparativa entre planificaciones

A lo largo del desarrollo del proyecto se han mantenido reuniones periódicas del alumno con el profesor José Massa, persona de contacto del Hospital Italiano de Buenos Aires, y con el tutor Simone Balocco. Desde un primer momento se ha definido el proyecto como la realización de una plataforma web con el visto bueno por parte del Hospital Italiano, tanto en el planteamiento inicial como en las posteriores reuniones.

Cabe destacar que el proyecto se ha planificado teniendo en cuenta una fecha de entrega marca en viernes 21 de diciembre de 2018. Esta fecha de entrega ha sido acordada entre el alumno y el tutor para poder tener un margen amplio para la revisión y corrección del documento de memoria y/o del código en caso de que fuese necesario. Es por este motivo, que la etapa del TFG va desde el día 16 de octubre de 2018 (fecha de aceptación de la idea) hasta el 21 de diciembre de 2018 (fecha acordada de “finalización” entre alumno y tutor). Teniendo en cuenta los límites de fechas del proyecto, juntamente con la primera propuesta (explicada en la sección 2.2), se realiza una planificación que puede ser visible en el diagrama de Gantt (ilustración 1). Aunque oficialmente la fecha de entrega está marcada a día 1 de febrero de 2019 por la facultad, el proyecto a fecha 21 de diciembre se presenta como finalizado.

En la reunión periódica entre el alumno y el profesor José Massa (por parte del Hospital Italiano) con fecha 23 de noviembre de 2018, se comunica que el Hospital Italiano requiere una aplicación móvil en lugar de una plataforma web. Teniendo en cuenta la fecha de esta comunicación (bastante avanzada en los plazos) y, teniendo en cuenta que desde un primer momento la idea de una plataforma web estaba aceptada por parte del hospital, se intenta analizar la situación en busca de una solución que pueda satisfacer las necesidades sin afectar en exceso a la planificación realizada.

Es por el motivo arriba comentado que, juntamente con la propuesta final (sección 2.3), se adjunta el diagrama de Gantt con la planificación modificada para incluir la solución de la app móvil. La principal diferencia entre la planificación inicial y la planificación final es el recorte de la asignación de horas o días de trabajo de algunas etapas. La etapa inicial de diseño, en concreto del



área privada, marcada inicialmente del 20 de noviembre de 2018 al 12 de diciembre de 2018, pasa a ser del 20 de noviembre de 2018 al 5 de diciembre de 2018. Este recorte permite que, la etapa de servidor, marcada inicialmente del 12 de diciembre de 2018 al 20 de diciembre de 2018, pase a ser del 6 de diciembre de 2018 al 13 de diciembre de 2018, recortando el plazo en un día más. Gracias a estos recortes y modificaciones, se permite introducir la etapa de generación de la app móvil (con su aprendizaje previo incluido), entre los días 14 de diciembre de 2018 y 20 de diciembre de 2018.

Es cierto que quizás puede parecer un cambio pequeño y no significativo pero, teniendo en cuenta los límites de fechas que estaban establecidos, cualquier tipo de recortes en el número de horas y/o días asignados a una etapa o tarea sí que puede resultar significativo y afectar al desarrollo del proyecto. Aún así el objetivo al rehacer la planificación siempre ha sido poder aportar una solución para incluir una app móvil sin afectar en exceso al estado actual del proyecto y a su desarrollo como la idea inicial estaba planteada.

## 3. Análisis de requerimientos

### 3.1 Primer análisis

Como se ha visto en el apartado de propuesta, la idea principal de la oferta del TFG que se puede ver en la referencia [1] requería de una plataforma en la cual el usuario pudiese subir una fotografía de algún lunar que tuviese en su cuerpo indicando la zona, así como algunos datos personales suyos. Tras esta subida, el algoritmo de análisis indicaría un resultado sobre el estado del lunar analizado.

El análisis del proyecto ha girado en torno a esta idea principal, pero al inicio del desarrollo se ha podido comprobar que se podían introducir algunas características más para hacer la plataforma más interesante, sobre todo aprovechando las funcionalidades de una plataforma web. Es por este motivo que se ha evolucionado la idea principal hacia una página web corporativa que sirva como punto de entrada informativo; y un área privada donde los usuarios (usuarios, doctores y admin) puedan desarrollar sus funciones, ya sean subir consultas en el caso de usuarios, responderlas en caso de doctores o gestionar todo tipo de usuarios en caso de los admin.

Al haber evolucionado la plataforma en esa línea, se aporta un grado más de dinamismo, pues el punto de entrada no es la funcionalidad del algoritmo en sí (es una página informativa) y, además, se aporta seguridad al encerrar la funcionalidad importante en un área privada accesible con usuario y contraseña (evitando que cualquier tipo de bot automático pueda hacer pruebas).

En los siguientes apartados se detalla cómo se ha realizado en análisis.

### 3.2 Requisitos funcionales

Cogiendo como referencia el segundo párrafo del apartado anterior, es posible observar que hay tres tipos de usuarios en la plataforma: usuario por defecto (usuario), doctores y administradores (admin). Además, debemos considerar el usuario “invitado”, en referencia a un usuario que visita la página pero no se ha identificado.

A continuación se describen los casos de uso de cada uno de ellos (es posible ver la descripción completa en el anexo, apartado 14.2):

### 3.2.1 Invitado

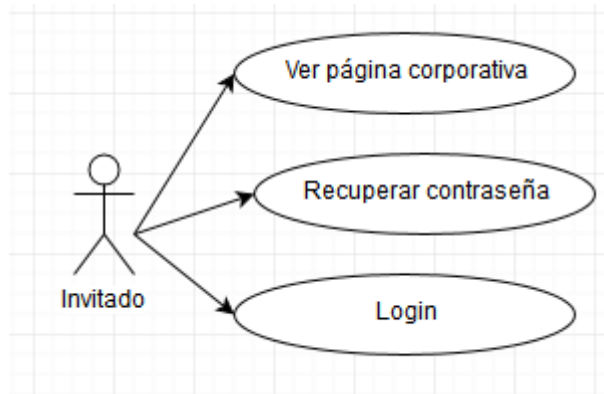


Ilustración 3: Casos de uso de invitado

Cuando un usuario visita la página pero no se identifica (es decir, no ha hecho login con un usuario y contraseña), tiene tres acciones posibles a realizar: navegar por la página corporativa (página inicial), hacer login o recuperar su contraseña.

Es posible ver más rápidamente los casos de uso del usuario invitado en la ilustración 3.

### 3.2.2 Usuario por defecto

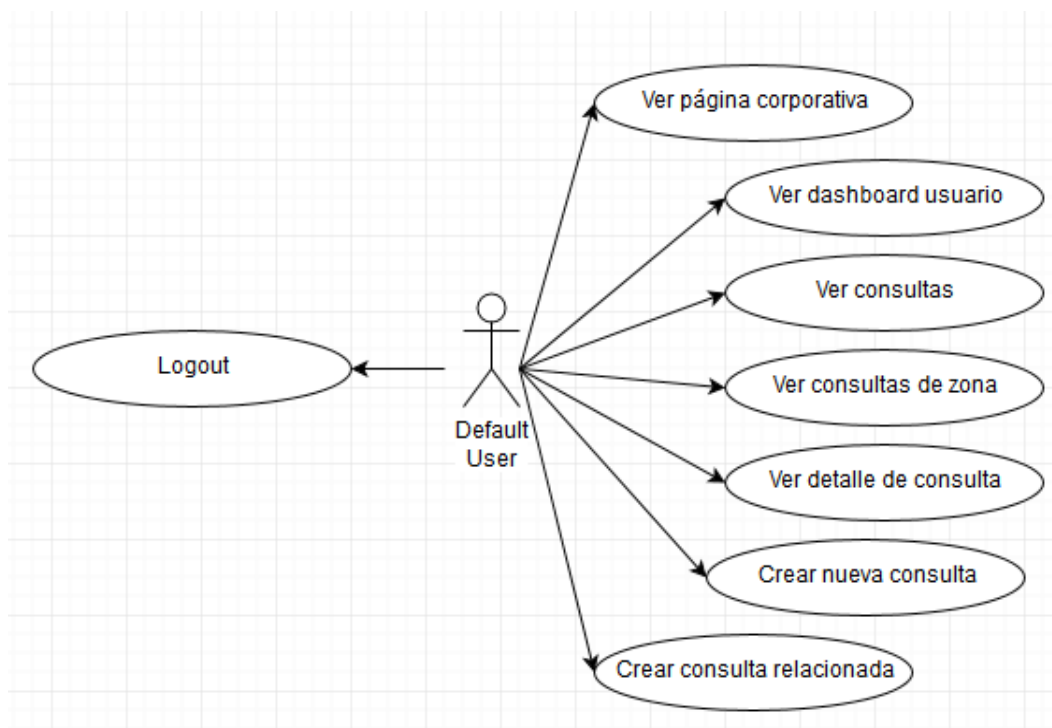


Ilustración 4: Casos de uso de usuario por defecto

Un usuario por defecto es el que ha accedido a la página y ha hecho login con unas credenciales de usuario (no de doctor ni de admin). Este tipo de usuario puede navegar por la página corporativa (accesible a cualquier usuario) y, como acciones propiamente concretas de un usuario por defecto logeado, puede hacer logout (desconectarse).

Además, dispone de una serie de acciones sobre las consultas:

- Acceder al panel de usuario por defecto.
- Ver todas sus consultas.
- Ver las consultas de una zona en concreto.
- Ver el detalle de una consulta.
- Crear una nueva consulta.
- Crear una consulta relacionada con una zona ya consultada.

En la ilustración 4 es posible ver los casos de uso del usuario por defecto.

### 3.2.3 Doctor

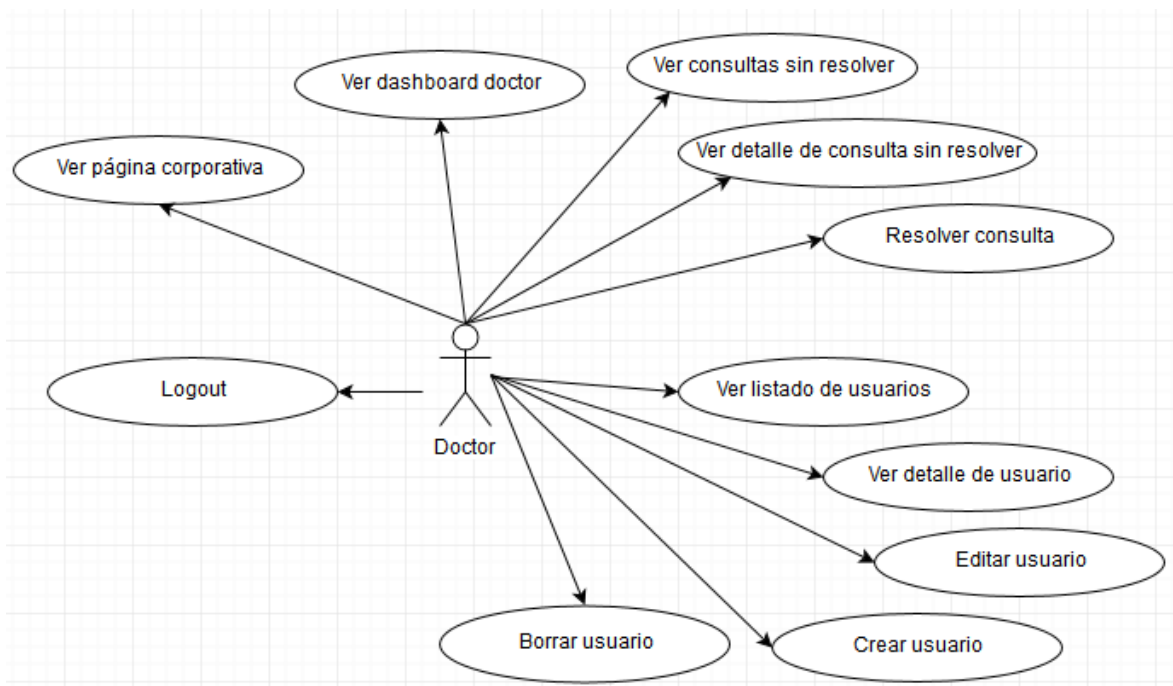


Ilustración 5: Casos de uso de doctor

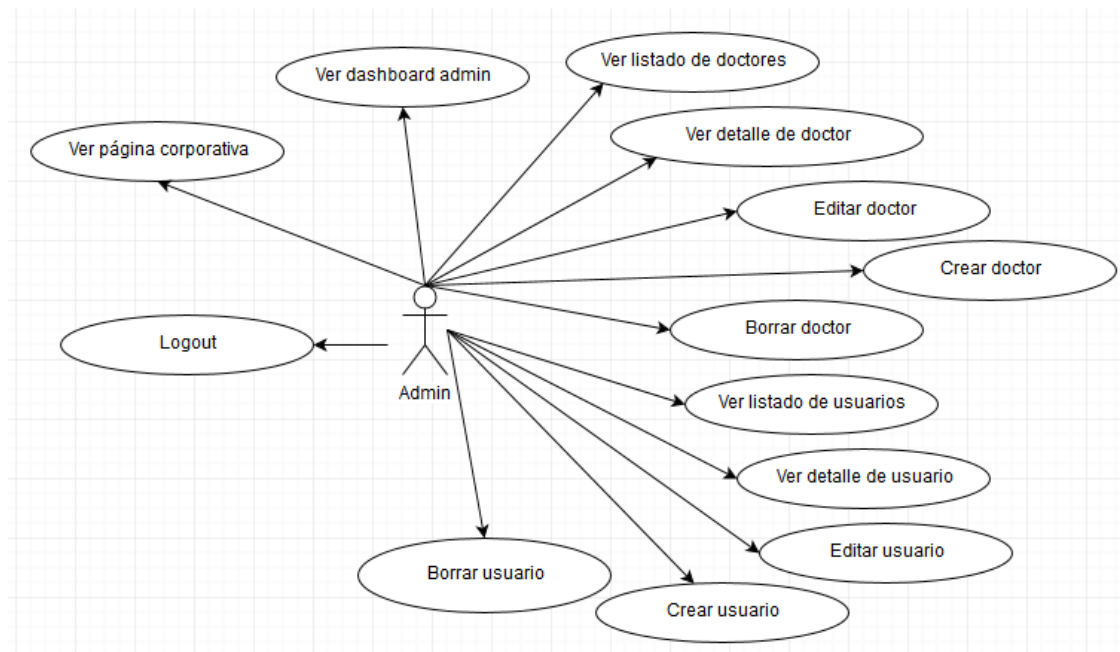
El usuario de tipo doctor es el que ha accedido a la página y se ha identificado con las credenciales de doctor. Las acciones posibles a realizar por parte de este usuario son muy parecidas al del usuario por defecto. El usuario doctor puede navegar por la página corporativa y hacer logout.

Como acciones propias de doctor, es posible listar las siguientes:

- Acceder al panel de doctor.
- Acciones sobre el listado de consultas:
  - Ver consultas sin resolver.
  - Ver detalle de una consulta sin resolver.
  - Resolver una consulta.
- Acciones sobre el listado de usuarios:
  - Ver listado de usuarios.
  - Ver detalle de un usuario.
  - Editar un usuario.
  - Crear un usuario.
  - Borrar un usuario.

En la ilustración 5 se pueden observar los casos de uso del usuario doctor.

### 3.2.4 Admin



*Ilustración 6: Casos de uso de administrador.*

Por último, el usuario de tipo administrador es el que ha visitado la página y se ha identificado con credenciales de tipo admin. Este usuario puede, del mismo modo que los demás, navegar por la página corporativa y hacer logout.

Como acciones concretas de tipo admin dispone de las siguientes:

- Ver el panel de administrador.
- Acciones sobre el listado de doctores:
  - Ver el listado de doctores.
  - Ver el detalle de un doctor.
  - Editar un doctor.
  - Crear un doctor.
  - Borrar un doctor.
- Acciones sobre el listado de usuarios:
  - Ver el listado de usuarios.
  - Ver el detalle de un usuario.
  - Editar un usuario.

- Crear un usuario.
- Borrar un usuario.

En la ilustración 6 es posible ver los casos de uso de administrador.

### 3.3 Requisitos no funcionales

Una vez se han comentado los requisitos funcionales, es posible adentrarse en los requisitos no funcionales que la plataforma requiere. Estos requisitos obedecen más a características que debería tener la herramienta web.

- **Interfaz responsive<sup>[11]</sup>:** En un primer momento, como el requerimiento era de una página web y, más adelante, apareció el requerimiento de una APP móvil, es importante que la interfaz sea responsive sencilla para que pueda adecuar el contenido al dispositivo desde el cual se está accediendo. De este modo se facilitaba una posible solución por si el requerimiento de APP no se pudiese conseguir, ya que desde dispositivos móviles el contenido sería visible de una manera cómoda.
- **Seguridad:** Como en la plataforma web habrán datos personales, se ha decidido adoptar alguna medida extra de seguridad. Estas medidas vienen facilitadas por el framework Laravel, que ya provee de muchos detalles por defecto en este sentido.

Uno de los factores que aporta Laravel (si se precisa utilizar, como es el caso) es proteger alguna de las rutas de la plataforma con un “middleware”, que no es más que una especie de “controlador” por el cual deben pasar todas las peticiones que vayan a las rutas que este middleware cubra. De este modo, dependiendo de cómo se configure el middleware, el acceso a esas rutas quedará restringido con las medidas que se consideren oportunas.

Teniendo en cuenta el aspecto anterior, para evitar posibles intentos masivos de subida de imágenes, se ha decidido crear un área privada para encerrar la aplicación en sí (formulario de subida de imagen). De este modo la estructura de la plataforma queda bien definida con un área exclusiva para los usuarios por defecto, a la cual no se puede acceder sin haber hecho “login”. Además, también se ha definido un área exclusiva para los usuarios doctores y, por último, para un usuario de tipo administrador, con tareas propias de gestión para doctores y/o administradores. El acceso a estas dos últimas áreas también es mediante un login previo.

Por último, se ha configurado que las imágenes que se usen en la aplicación en las consultas de los usuarios se almacenen en un directorio privado (a diferencia del directorio por defecto, que es el directorio “public”) con un nombre aleatorio, consistente en una serie de caracteres (que por sí solos no identifican al usuario). Gracias a este cambio las imágenes no son accesibles de manera fácil, ya que para acceder se precisa de una URL protegida,

accesible únicamente por el usuario que ha subido esa imagen (otro usuario por defecto no podría verla).

- **Escalabilidad:** Hay algunos aspectos que no entran dentro del ámbito del proyecto (como el algoritmo de análisis del lunar). Sin embargo se requiere que la plataforma esté finalizada de un modo que sea fácil poder introducir mejoras en el futuro si algún alumno quiere ampliar este TFG.
- **Uso de scripts Python:** Aunque la plataforma sea desarrollada principalmente con el framework Laravel, es muy importante implementar la posibilidad de ejecutar scripts de Python dentro de la aplicación. El motivo de tener en cuenta esta característica es que el algoritmo que analiza el lunar de la imagen que el usuario suba será un script Python (perteneciente a otro futuro TFG).

Teniendo en cuenta este requisito, en la plataforma se ha implementado la ejecución de un pequeño script Python cada vez que se sube una imagen, el cual cambia los colores de la imagen subida. De este modo queda demostrado la posibilidad de la plataforma a ejecutar el futuro algoritmo Python cuando se desarrolle.



## 4. Diseño del modelo de datos

### 4.1 Modelo entidad/relación

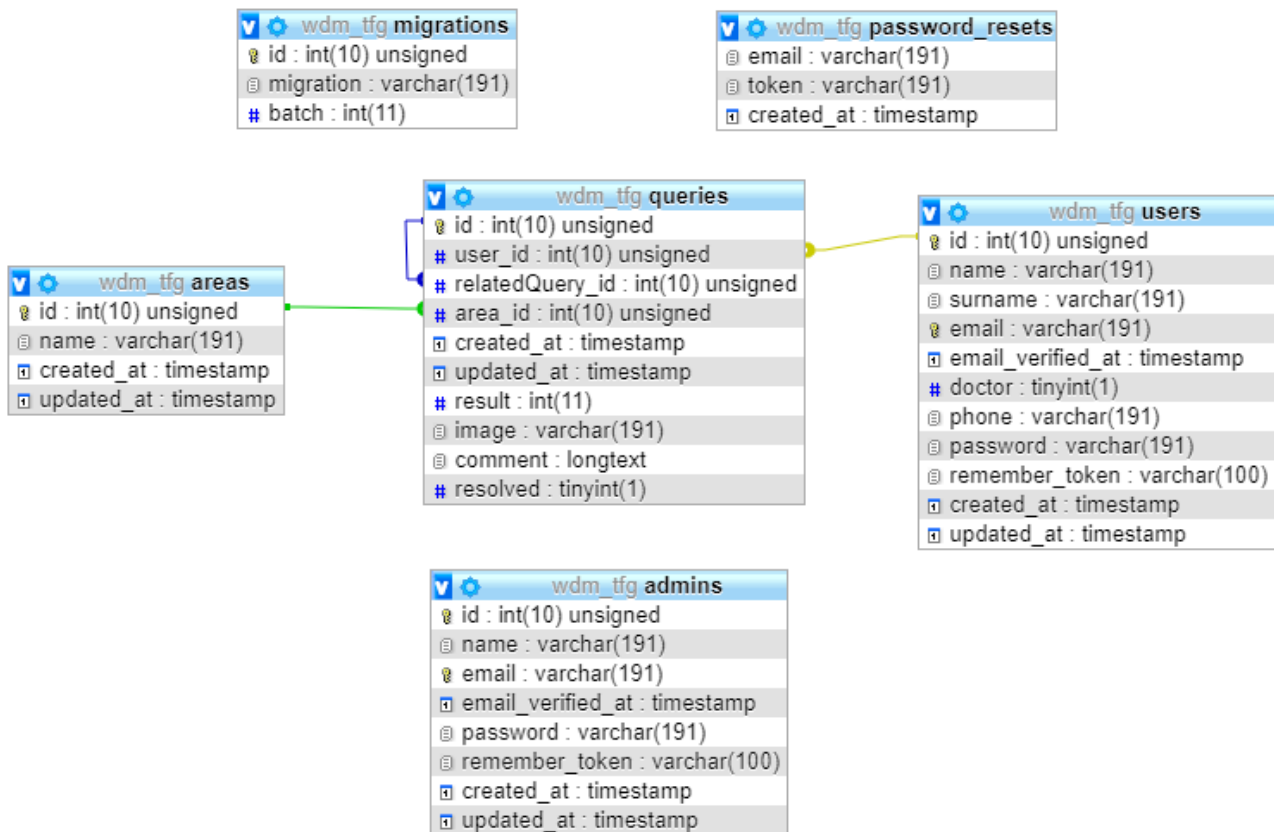


Ilustración 7: Modelo entidad/relación

En la ilustración 7 podemos ver el modelo entidad/relación extraído directamente del cliente de BBDD PHPMyadmin<sup>[13]</sup>. En dicho modelo es posible ver las tablas propias de la que utiliza la plataforma web así como las relaciones que se han establecido.

## 5. Tecnología del proyecto

### 5.1 Servidor

Para hablar de la tecnología utilizada en el proyecto, conviene diferenciar entre la fase de desarrollo y el entorno de producción en sí, ya que en cada fase la configuración no ha sido exactamente la misma. Sí que es cierto que los componentes instalados son los mismos, pues son los requeridos para que la plataforma funcione, pero el modo de instalar dichos componentes ha sido distinto.



*Ilustración 8: Estructura de componentes de WampServer*

Por un lado, durante la fase de desarrollo (entorno local), se ha instalado directamente WampServer<sup>[14]</sup>, consistente en el entorno WAMP<sup>[15]</sup>. WAMP es el acrónimo de Windows (Sistema Operativo), Apache (servidor web), MySQL (base de datos) y PHP, tal y como se puede ver en la ilustración 8. Y, a su vez, WampServer es un software que instala y configura todos los componentes de un entorno WAMP. El resultado de esta instalación es la posibilidad de poder desarrollar de manera local un entorno web completamente accesible localmente mediante localhost.

Es importante mencionar el entorno de desarrollo, pero la explicación de tecnologías y de la configuración de todo el entorno (en el anexo) corresponderá al entorno de producción, donde la plataforma actualmente se encuentra alojada y funcionando, accesible online.

En el caso del entorno de producción el servidor utilizado ha sido Amazon WebServices<sup>[16]</sup>, consistente en una máquina virtual online EC2<sup>[17]</sup>. Esta máquina virtual está accesible de manera externa mediante SSH<sup>[18]</sup>. El sistema operativo de la máquina virtual es Ubuntu<sup>[19]</sup>, en su versión Ubuntu 18.04.1 LTS.

## 5.2 Servidor Web - Apache

Una vez ya está el servidor ya está listo y funcionando, se debe configurar para otorgar acceso al mismo vía web, ya que actualmente la conexión era por SSH utilizando línea de comandos. Para el proyecto es necesario poder visualizar la plataforma web que alojará el servidor.



*Ilustración 9: Apache HTTP Server*

Para ello se utilizará Apache<sup>[20]</sup>, que es un servidor web de código abierto. Esto permite poder realizar cualquier tipo de modificación en su código para poder adecuarlo al proyecto que se está realizando. Del mismo modo, tiene una comunidad de desarrolladores muy grande dedicada y, por este motivo, recibe actualizaciones de forma constante.

La configuración de Apache es muy sencilla (se podrá ver en el Anexo de configuración) y su instalación, gratuita. Al utilizar Apache es posible beneficiarse de los muchos componentes que realiza la comunidad. Además, este servidor web es compatible con muchos lenguajes de programación web.

La versión instalada es la versión Apache/2.4.29.

## 5.3 Lenguaje de programación web – PHP



*Ilustración 10: Lenguaje de programación web PHP*

PHP<sup>[21]</sup> es un lenguaje de programación que ha tenido mucho éxito como lenguaje de desarrollo web. PHP permite introducir scripts entre el código HTML que se ejecutará del lado del servidor. Al ejecutarse el script tan solo se ve el resultado como código HTML, por lo tanto el usuario no sabe realmente lo que hay detrás.

Al ser un lenguaje muy extendido, la mayoría de servidores o hostings aceptan proyectos realizados con este lenguaje. Esto permite que cualquier tipo de migración pueda realizarse de una manera más cómoda puesto que al hacerlo no deberían presentarse incompatibilidades relacionadas, más allá de adecuarse a la versión que se utiliza. En el caso del proyecto presente, es la versión 7.2.10.

Este lenguaje es muy importante en el proyecto, ya que la lógica de la plataforma está implementada en PHP. Esto quiere decir que detrás de todo lo visible al acceder a la web, hay mucho código de PHP que dice a la aplicación lo que debe hacer en cada caso. Es posible ver muchos ejemplos si se piensa en la lógica para guardar un usuario en el sistema o las verificaciones del usuario que tiene iniciada sesión.

Aún así, aunque sería completamente posible desarrollar la plataforma web utilizando código original de PHP para toda su lógica, se podrá ver más adelante (en la sección 5.5) cómo se ha utilizado un método que facilitará mucho el desarrollo.

## 5.4 Gestor de dependencias – Composer



*Ilustración 11: Gestor de dependencias Composer*

Cuando se utiliza un lenguaje de programación en concreto, sobre todo aquellos que están muy extendidos como PHP, es muy probable que la gran comunidad que tiene detrás desarrolle componentes, paquetes o librerías que ayudan a realizar alguna función en concreto. Lo bueno de estas pequeñas “extensiones” es que la manera en la que permiten hacer las acciones normalmente es mucho más sencilla y ágil que utilizando directamente el lenguaje PHP de cero. Esto no quiere decir que estos paquetes o librerías no estén escritos en PHP, al contrario, significa que otros desarrolladores han implementado un código PHP reduce la dificultad de muchas acciones a fáciles comandos o funciones.

El principal problema que surge ante este hecho es que, a la hora de querer utilizar cierto paquete o librería, sería necesario acudir al lugar donde están almacenados (normalmente una web), registrarse si se da el caso, incluirlo en el proyecto, controlar que lo que se quiere usar no requiere ninguna otra librería o paquete de terceros, el cual no se tiene por qué conocer...

Para solucionar esto existen los gestores de dependencias. Para PHP la solución nace con Composer<sup>[22]</sup>. Composer es un gestor de dependencias de PHP, y permite poder instalar paquetes y librerías muy fácilmente. Normalmente, desde el terminal, con un simple comando, se puede instalar la extensión que se precise. Además, Composer hará una búsqueda e instalará todos los otros paquetes o librerías de terceros que sean requeridos para instalar la extensión deseada. De esta manera el usuario no debe preocuparse más que saber qué quiere instalar, sin importar todas las dependencias de esa extensión ni la versión actual que esté disponible.

Cabe destacar que Composer instala todas las extensiones que el usuario requiera a nivel de proyecto. Esto quiere decir que si se crea otro proyecto no estarán disponibles todas estas extensiones, por lo que cada proyecto está configurado de forma independiente a nivel de paquetes y librerías.

La versión de Composer instalada es la versión 1.8.0.

## 5.5 Framework PHP – Laravel



*Ilustración 12: Framework de PHP Laravel*

Esta sección del apartado de tecnología es, quizás, la más importante de todas. Es la más importante porque la elección de la tecnología usada como framework de PHP, en este caso, hace que el desarrollo se realice de un modo u otro. Durante la fase de análisis de este TFG, el alumno tomó la decisión de realizar el proyecto con Laravel<sup>[23]</sup>. Analizando los frameworks de PHP, quizás el que más éxito o “auge” tiene actualmente es Symfony<sup>[24]</sup> ya que es quizás el más extendido, pero Laravel también tiene detrás una comunidad muy grande que aporta su ayuda en forma de tutoriales y componentes, haciendo que no se quede atrás con respecto de Symfony.

Esta elección tenía su apoyo también en el hecho que la curva de aprendizaje de Laravel era mucho más suave que la de Symfony y, junto con el plazo de tiempo presente para el TFG, la primera opción supondría un impacto menor en cuanto a horas de dedicación. Esto permitiría al alumno poder centrarse más en el desarrollo del proyecto y en las funcionalidades, ya que podría dedicar más tiempo al desarrollo con respecto al aprendizaje de la tecnología.

Aún así, ese hecho no quiere decir que Laravel no vaya a tener el mismo rendimiento que Symfony ya que el primero, por ejemplo, incluye muchos componentes y/o librerías directamente del mismo framework Symfony. Simplemente para el tipo de proyecto que se realiza Laravel cumple a la perfección y ayuda en los plazos de aprendizaje, siendo una herramienta muy completa.

Por último, hay que mirar al futuro una vez se haya finalizado el TFG. Quizás otro alumno quiera centrar su TFG en ampliar el que se está explicando y, si se da el caso de que no tiene experiencia con frameworks, es más sencillo que pueda llegar a aprender y a utilizar Laravel que Symfony. Además, Laravel dispone de una comunidad detrás muy grande que resuelven muchas dudas y dispone de una documentación<sup>[25]</sup> que está muy bien estructurada, con un contenido muy claro y preciso.

La versión de Laravel utilizada en el proyecto es la versión 5.7.16.

## 5.6 Framework HTML & CSS - Bootstrap



*Ilustración 13: Framework HTML&CSS  
Bootstrap*

Para ayudar al desarrollo de toda la parte visual y estética de la plataforma web se ha utilizado un framework HTML<sup>[26]</sup> & CSS<sup>[27]</sup> (también usa JavaScript<sup>[28]</sup>) llamado Bootstrap<sup>[29]</sup>.

Bootstrap ayuda al desarrollador aportando una serie de ficheros de scripts de JavaScripts y una hoja de estilos CSS muy bien hechos y elaborados que permiten introducir en la página muchos componentes tanto simples (botones e inputs) como más complejos (como menús desplegables) simplemente copiando el código que se puede ver en la documentación<sup>[30]</sup> de su página. Esta documentación está estructurada y separada por secciones, de modo que el usuario puede buscar fácilmente cualquier tipo de característica que desee introducir para ver ejemplos y distintos modelos.

Por último, Bootstrap ofrece un layout en concreto para sus páginas que facilita la inclusión de los elementos HTML en cualquier parte. Esta acción de colocar elementos en según que sitios a menudo es una tarea un poco tediosa para el desarrollador, sobre todo vigilando que ninguno de los demás elementos se descoloquen. En este sentido, el framework es de tipo responsive<sup>[11]</sup> por lo que, en mayor medida, el contenido de la plataforma web se adaptará al tamaño del dispositivo donde se está visualizando de una manera cómoda y accesible.

Aunque hay diseños más básicos o “minimalistas” y otros más complejos, Bootstrap ha ido adquiriendo un muchos adeptos que recurren a este framework para el desarrollo de su parte visual, y cualquier tipo de diseño realizado de este modo aporta a la web un acabado actual.

## 5.7 Control de versiones – Git/Github



*Ilustración 14: Control de versiones Git*



*Ilustración 15: Servicio público de control de versiones GitHub*

Hace algunos años, en el desarrollo web, se trabajaba de un modo más directo, ya que para realizar alguna modificación de los archivos de una plataforma web, simplemente se editaban a nivel local y, mediante cualquier cliente FTP, se subían al servidor para poder observar los cambios. Esta metodología de trabajo al final derivaba en que si aparecía algún problema no se podía disponer de un historial de cambios completo donde se pudiese ver qué se ha modificado y quién lo ha hecho. Además, cualquier tipo de nueva implementación requería de una subida como tal al servidor.

Para solucionar este problema se dispone de una herramienta muy útil: el control de versiones. El control de versiones no es más que una herramienta que provee de un lugar donde subir el proyecto (repositorio) aportando una serie de funcionalidades. Una vez se ha subido el proyecto, cada vez que se requiera una modificación se deben realizar unas acciones llamadas “commits”, que no es más que una especie de “etiquetado” de las modificaciones que se han hecho con una breve descripción. Además, la propia herramienta refleja los archivos modificados, qué líneas en concreto se han modificado así como el autor de dichos cambios. Estos commits se



reflejan a nivel local hasta que se realiza una acción llamada push, consistente en subir al repositorio ese commit.

La ventaja de trabajar con esta herramienta es que es posible volver a un estado anterior (a un commit anterior) muy fácilmente, revocando todos los cambios de forma automática.

Trabajar con un control de versiones tiene muchas más funcionalidades, como el trabajo en ramas (generar una “copia” del estado actual del repositorio desde la cual continuar y realizar los cambios). Esto permite tener mayor seguridad al no estar trabajando directamente sobre el código “original”. Al finalizar las modificaciones, por ejemplo, se pueden mezclar los cambios de la rama con los del código original, controlando los cambios.

Por último, soluciona el tan extendido problema a la hora de trabajar en grupos de varios desarrolladores, pues siempre existe el problema de que uno puede estar pisando el trabajo de otro sin darse cuenta. El control de versiones tiene esto en cuenta y, cuando va a suceder, genera un warning para que el desarrollador pueda revisar qué parte del código está en conflicto con algún cambio que ha hecho otro desarrollador al mismo tiempo.

En el caso del proyecto actual, el control de versiones utilizado es Git<sup>[31]</sup>. Como servicio público de Git, se ha usado GitHub<sup>[32]</sup>, que no es más que la plataforma donde se alojan los repositorios que, en este caso, usan Git.

## 5.8 Editor de texto/IDE – Visual Studio Code

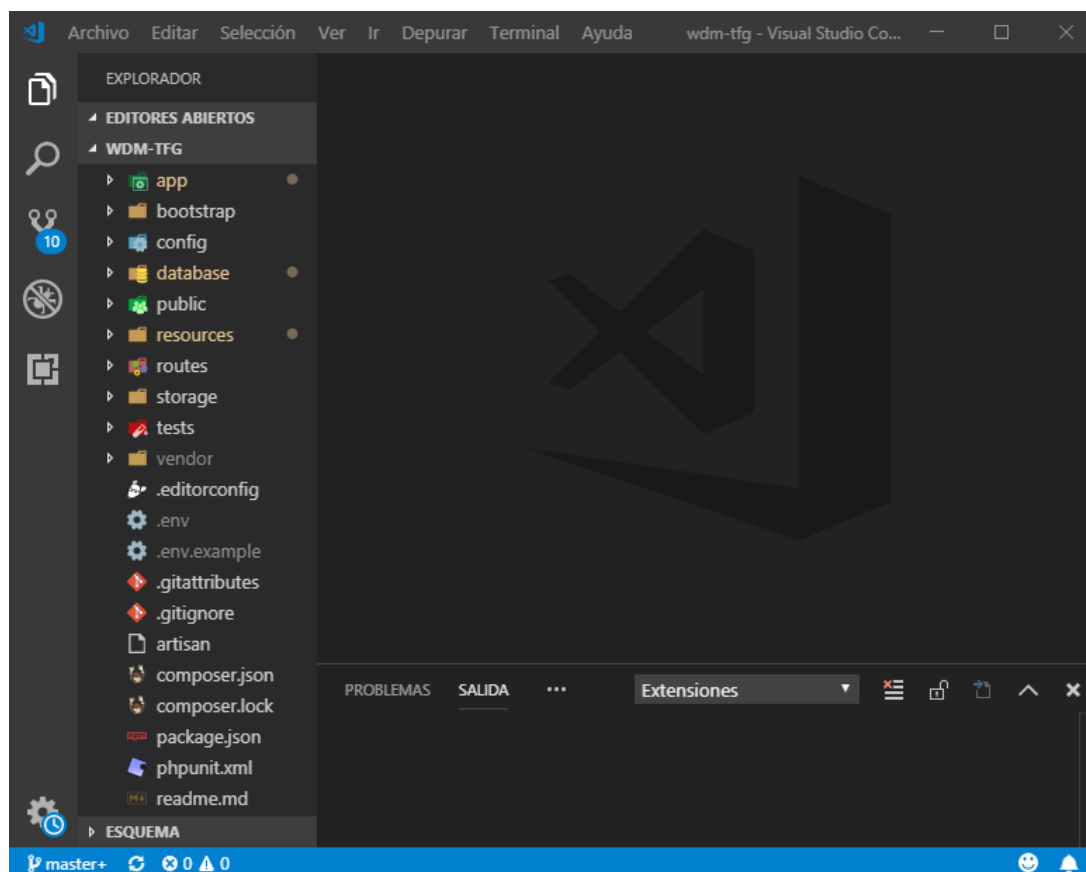


*Ilustración 16: Editor de texto Visual Studio Code*

Para escribir el código de la plataforma se precisa de un editor de texto o de un IDE (entorno de desarrollo). Un editor de texto podría ser el típico bloc de notas de Windows (se puede utilizar para programar también) pero, normalmente, se usan otros editores de texto que aporten algunas funcionalidades extra a nivel de programación. Funcionalidades que faciliten la escritura de código, como autocompletado o sugerencias. Un IDE es similar, pero aporta muchísimas más

funcionalidades y es más complejo. Suelen ser más pesados (en cuanto a tamaño) y suelen consumir más recursos.

En el proyecto actual se ha elegido un editor de texto, en este caso, Visual Studio Code<sup>[33]</sup>, de Microsoft<sup>[34]</sup>. Este editor de texto ha sufrido muchos cambios actualmente, habiendo sido renovado por parte de Microsoft para adecuarlo a los tiempos actuales. Además, la descarga y el uso de este editor son gratuitos.



*Ilustración 17: Captura del editor de texto Visual Studio Code*

Visual Studio Code ofrece muchas ayudas tanto para el desarrollo de PHP como para Laravel, directamente. Tiene el punto positivo de tener un apartado de extensiones con un buscador que permite encontrar extensiones de todo tipo, más allá de las que de por sí ofrece, como paquetes de idiomas hasta distintos paquetes que realiza la comunidad para sugerencias de PHP, Laravel...etc. Tiene la posibilidad también de poder ver directamente un terminal (tanto Shell como Bash) para ejecutar comandos desde el propio editor.

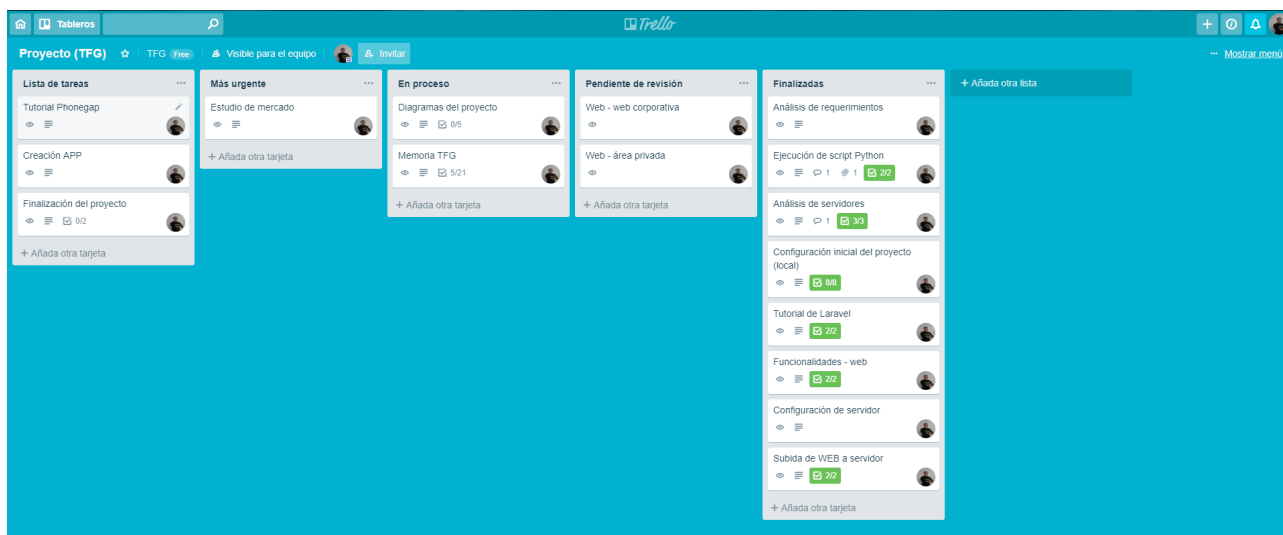
## 5.9 Administración y gestión – Trello



*Ilustración 18: Software de administración de proyectos Trello*

El software descrito en esta sección no es esencial para el trabajo (es posible realizar el proyecto sin usarlo), pero su uso es muy importante cuando el usuario se acostumbra a gestionar un proyecto con él.

En el caso del alumno, ha escogido usar la plataforma Trello<sup>[35]</sup>. Esta plataforma se utiliza para gestionar todas las tareas de un proyecto. Cuando se gestiona un proyecto, es posible crear diferentes estados en los que se puede encontrar una tarea (pendiente, en proceso...). A su vez, permite crear “tarjetas” a modo de tareas, pudiendo establecer título, descripción, listado de subtareas... De este modo, de una forma muy rápida y visual, es posible ver el estado en el que se encuentran todas las tareas a realizar.



*Ilustración 19: Captura del TFG “WDM” en Trello*

La forma de pasar una tarea de un estado a otro es muy intuitiva, ya que simplemente haciendo “drag&drop” de un estado a otro la tarjeta quedará establecida en ese estado.

Como se ha comentado, es posible realizar el proyecto sin el uso de esta herramienta, pero el hecho de disponer de ella y utilizarla correctamente ofrece una ayuda muy grande para no olvidarse de ningún punto durante el transcurso de cualquier tipo de trabajo a realizar.

## 6. El framework Laravel

### 6.1 Componentes de Laravel

Tal y como se ha comentado en apartado de tecnologías (apartado 5), uno de los aspectos importantes de la preparación del proyecto es decidir cómo se debe realizar. En este caso la elección ha sido utilizar el framework PHP Laravel. Habían otras opciones a considerar, ya que se podría haber usado cualquier framework PHP como Symfony, tal y como se ha comentado; buscar otra clase de framework (JavaScript por ejemplo) como Angular<sup>[36]</sup> u optar por realizar el proyecto directamente con lenguajes “base”, como HTML, CSS y PHP.

Durante esa fase de análisis para ver cómo realizar el proyecto, el tutor Simone Balocco dio absoluta libertad en cuanto a elección, pero recomendó usar algún framework. Al alumno también le parecía bien la idea y, aprovechando sus conocimientos previos de PHP, optó por ese tipo de framework. La elección de Laravel sobre Symfony, tal y como se ha comentado en el apartado 5, corresponde más a la curva de aprendizaje más suavizada en el primero que en el segundo, además de ver un poco hacia el futuro por si algún otro alumno centra su TFG en extender el que se está explicando y no tiene conocimiento de frameworks: le será más fácil aprender Laravel que Symfony.

¿Hubiese sido mala elección no usar ningún framework y usar lenguajes “base”? No tendría nada de malo, y el proyecto se hubiese podido realizar igual. Pero una de las características de Laravel (y de los framework en general) es que facilitan el desarrollo al usuario. Simplemente al desplegar un proyecto con Laravel, el estado en el que se encuentra ese proyecto, ya supone algunas horas de dedicación si no se usara el framework y se hiciera con lenguajes base.

Esto es así porque el framework, al crear un proyecto, ya genera una estructura de carpetas determinada, bien organizada, separando cada una de las partes de la aplicación. Además, genera un contenido inicial como puede ser el enrutamiento inicial, pantalla principal con login y registro de usuarios (si se requiere)...etc. Como es posible observar, realizar desde cero una implementación como la que ofrece Laravel desde un principio requerirá una dedicación considerable.

Pero no solo se encarga de proporcionar un inicio para la aplicación, si no que si el usuario sigue la línea de desarrollo del framework adquirirá una serie de ventajas. Es decir, si a medida que desarrolla lo hace siguiendo los patrones logrará realizar las cosas más rápido y más seguro. Todos los componentes y archivos propios del framework no están más que para facilitar al desarrollador el implementar una función ya que enmascara toda la lógica necesaria (que ya está implementada) bajo una o dos acciones concretas a realizar por parte del usuario. Por poner un ejemplo, el generar url más estéticas.

## 6.2 Sistema de plantillas Blade

Laravel ofrece en su framework un sistema de plantillas basado en PHP llamado Laravel (de hecho, su extensión es .blade.php) llamado Blade<sup>[37]</sup>

```

1  <html>
2  <head>
3  <title>Problema</title>
4  </head>
5  <body>
6      <?php
7          if ($a > $b) {
8              echo "a es mayor que b";
9          } elseif ($a == $b) {
10             echo "a es igual que b";
11          } else {
12             echo "a es menor que b";
13          }
14      ?>
15  </body>
16  </html>

```

*Ilustración 20: Ejemplo de sentencia IF en PHP, incrustado en lenguaje HTML*

Utilizando PHP es posible introducir código de este lenguaje entre el lenguaje HTML, siempre enmarcado entre las llaves de PHP (<?php ?>) tal y como se ve en la ilustración 20. En esta ilustración se puede ver el ejemplo de una sentencia if. Este ejemplo es sencillo, pero conjuntar el lenguaje HTML dentro de un “echo” de PHP no es muy estético. Otra opción es ir abriendo y cerrando las etiquetas PHP cada vez que se introduzca código HTML, pero esto no es muy cómodo.

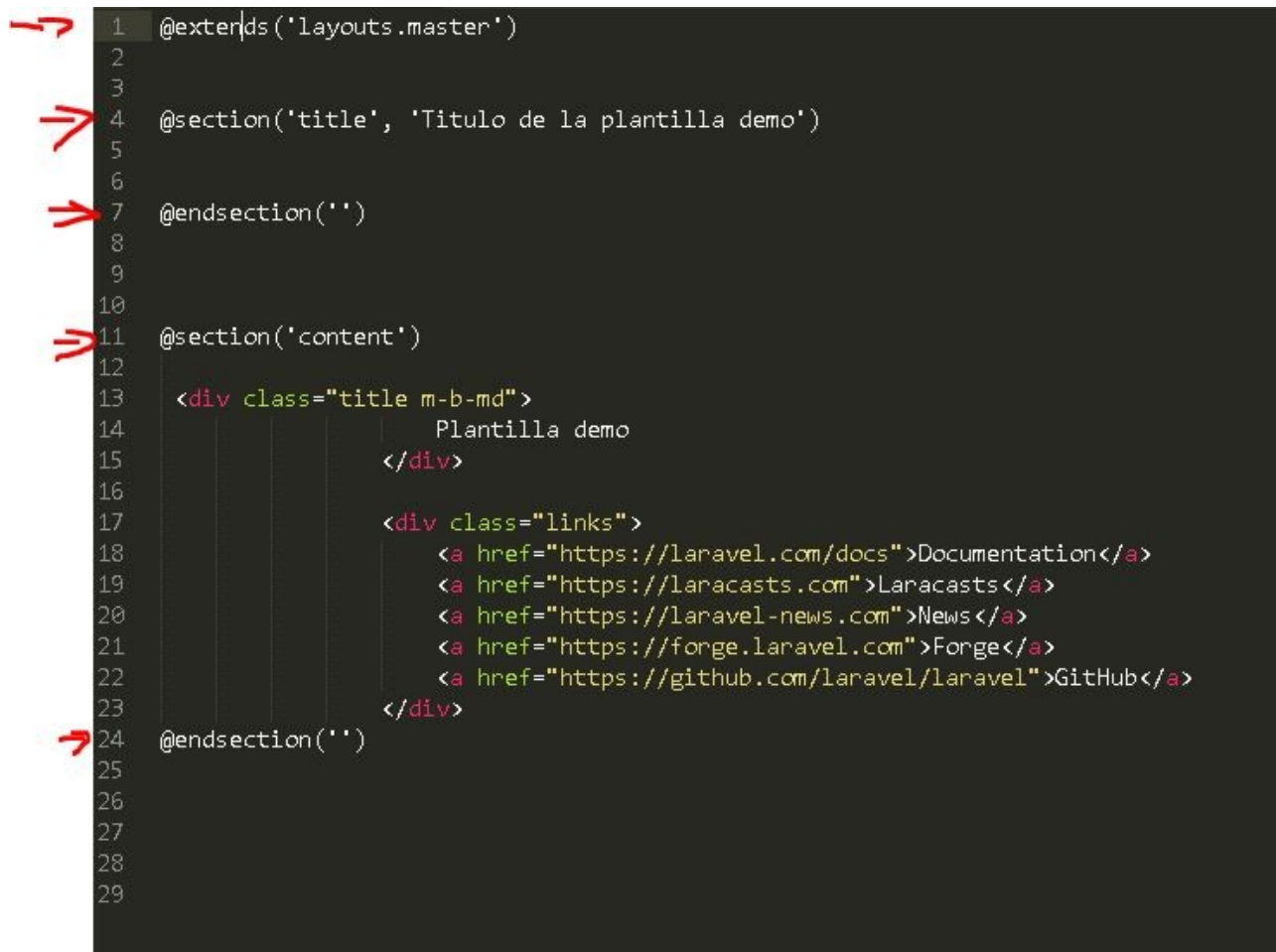
```

@if ($message = Session::get('success'))
<div class="alert alert-success alert-block">
    <button type="button" class="close" data-dismiss="alert">&times;</button>
    <h4>Success</h4>
    @if(is_array($message))
        @foreach ($message as $m)
            {{ $m }}
        @endforeach
    @else
        {{ $message }}
    @endif
</div>
@endif

```

*Ilustración 21: Ejemplo de sentencia IF usando directivas Blade en Laravel*

En la ilustración 21 se puede ver la solución que ofrece Laravel: directivas Blade. En este caso, sin abrir etiquetas de PHP, simplemente con “@if” ya se indica que allí habrá una sentencia if usando PHP. Dentro de cada sección se puede usar directamente código HTML e, incluso, encerrar variables de PHP entre “{{ }}”. Realizarlo de esta manera es mucho más sencillo que como se ha explicado un poco más arriba. En la documentación de Laravel se pueden ver todo tipo de directivas que facilitan este tipo de acciones.



```
1 @extends('layouts.master')
2
3
4 @section('title', 'Titulo de la plantilla demo')
5
6
7 @endsection('')
8
9
10
11 @section('content')
12
13     <div class="title m-b-md">
14         Plantilla demo
15     </div>
16
17     <div class="links">
18         <a href="https://laravel.com/docs">Documentation</a>
19         <a href="https://laracasts.com">Laracasts</a>
20         <a href="https://laravel-news.com">News</a>
21         <a href="https://forge.laravel.com">Forge</a>
22         <a href="https://github.com/laravel/laravel">GitHub</a>
23     </div>
24 @endsection('')
25
26
27
28
29
```

Ilustración 22: Captura de directivas extends y section de Blade en Laravel

Otro de los puntos interesantes que ofrecen las plantillas Blade evita al usuario repetir código. Por ejemplo, si todas las páginas disponen de un header o un footer en concreto, permite definir una archivo “layout” donde se introduce el código de esa parte a repetir. O, incluso, definir todo un layout formado por header, footer y barras laterales, definiendo una sección central para el código a introducir. A partir de ese momento, las páginas que sigan este layout tan solo deberán extender del layout de ejemplo para incluirlo. Acto seguido, con la directiva extends se incluirá ese layout en concreto y, con la directiva section, se puede establecer el contenido que irá en la sección central del layout. Es posible ver un ejemplo en la ilustración 22.

Utilizar todas las funcionalidades que el framework ofrece al usuario hacen que el desarrollo sea más sencillo y rápido. Además, mejora en un grado muy alto la legibilidad del código implementado y evita la repetición.

## 6.3 Eloquent ORM

Otra de las características de Laravel para facilitar al desarrollador el código a introducir es Eloquent ORM<sup>[38]</sup> (ORM son las siglas de mapeo objeto-relacional). Un ORM permite convertir los datos de los objetos del código que el usuario implementa en Laravel a un formato correcto para guardarlos en la base de datos que se utilice. Esta conversión de formato es “invisible” para el usuario, ya que no ve exactamente cómo Laravel transforma e introduce en la base de datos la información en concreto.

```
class PostController extends Controller
{
    public function store(Request $r)
    {
        $data = $r->validate([
            'title' => 'required|string',
            'content' => 'required|string',
        ]);

        Post::saveDraft($data['title'], $data['content']);

        return redirect()->route('posts.index');
    }

    public function publish(Post $post)
    {
        $post->publish();

        return back();
    }
}
```

*Ilustración 23: Ejemplo de Controlador con Eloquent*

```

class Post extends Model
{
    public static function saveDraft($title, $content)
    {
        $post = new Post();
        $post->title = $title;
        $post->content = $content;
        $post->save();
    }

    public function publish()
    {
        $this->published_at = now();
        $this->save();
    }
}

```

*Ilustración 24: Ejemplo de modelo con Eloquent*

Partiendo de este hecho, se vuelve a reincidir en la característica de que con el framework, mediante unas líneas de código sencillas y más elegantes, se pueden realizar acciones que requerirían de más trabajo, tal y como se ve en la ilustración 23 y 24. En la ilustración 24 se ve la implementación de cómo se guarda la información en la base de datos; y en la ilustración 23 se ve el uso del método “saveDraft” del ejemplo mediante el uso del modelo.

```

>>> $game = new Game;
=> App\Game {#675}
>>> $game->title = "Ninja Gaiden";
=> "Ninja Gaiden"
>>> $game->publisher = "Koei Temco";
=> "Koei Temco"
>>> $game->releasedate = "1988";
=> "1988"
>>> $game->save();
=> true
>>> .....

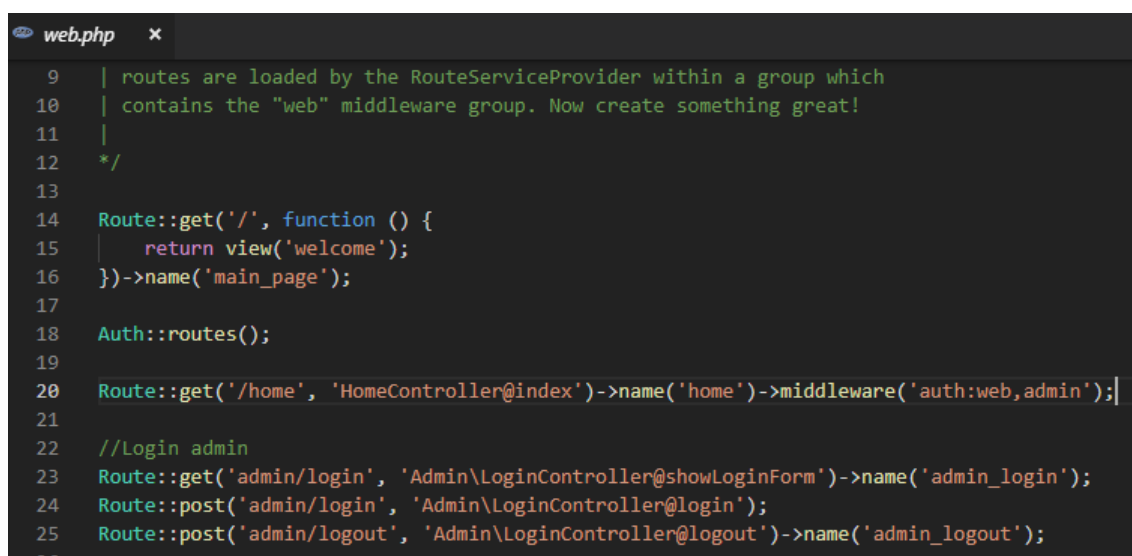
```

*Ilustración 25: Ejemplo de uso de Tinker*

Eloquent ORM de Laravel también ofrece una herramienta de interacción con la base de datos ejecutada en terminal, como se puede ver en la ilustración 25. Haciendo uso de este método se pueden realizar consultas a la base de datos rápidamente así como realizar modificaciones de la misma.



## 6.4 Routing y middlewares



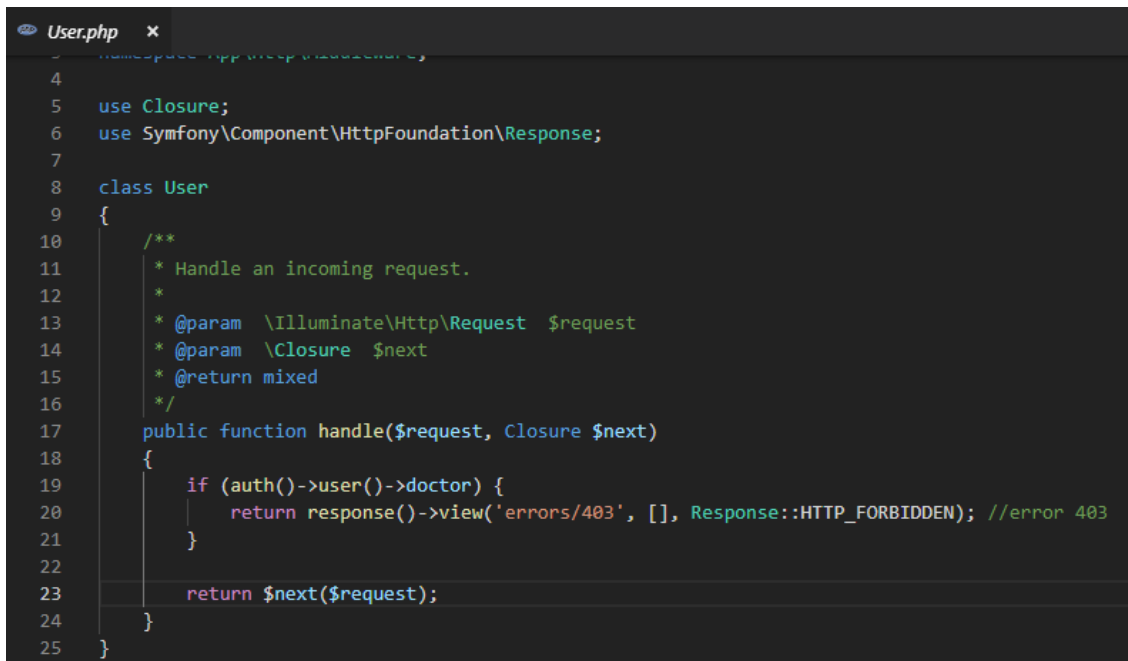
```
web.php x
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 | */
13
14 Route::get('/', function () {
15     return view('welcome');
16 }->name('main_page');
17
18 Auth::routes();
19
20 Route::get('/home', 'HomeController@index')->name('home')->middleware('auth:web,admin');|
21
22 //Login admin
23 Route::get('admin/login', 'Admin\LoginController@showLoginForm')->name('admin_login');
24 Route::post('admin/login', 'Admin\LoginController@login');
25 Route::post('admin/logout', 'Admin\LoginController@logout')->name('admin_logout');
26
```

*Ilustración 26: Ejemplo de un archivo de rutas en Laravel*

El control de rutas (routing<sup>[39]</sup>) mediante Laravel es muy sencillo. Al crear el proyecto el propio framework ya ofrece una sección en concreto para controlar esta funcionalidad. Consiste simplemente en un archivo .php en el que se definen las reglas de las rutas de la plataforma web que el usuario está implementando.

Como se puede observar en la ilustración 26, definir una ruta es tan sencillo como una línea de código (pueden ser varias simplemente para una mejor presentación de código). Para definir una ruta tan solo es necesario indicar el método (get, post...), la ruta en concreto (lo que se verá en la barra de dirección del navegador), el método y controlador que controlará la lógica de esa ruta y diversos parámetros opcionales, tales como el nombre que el usuario puede dar a la ruta, parámetros...etc.

Los archivos de ruta pueden dividirse por secciones (por ejemplo, rutas para admin, doctor...) para una mayor organización y visualización.



```

1  namespace App\Http\Middleware;
2
3
4
5  use Closure;
6  use Symfony\Component\HttpFoundation\Response;
7
8  class User
9  {
10     /**
11      * Handle an incoming request.
12      *
13      * @param \Illuminate\Http\Request $request
14      * @param \Closure $next
15      * @return mixed
16      */
17     public function handle($request, Closure $next)
18     {
19         if (auth()->user()->doctor) {
20             return response()->view('errors/403', [], Response::HTTP_FORBIDDEN); //error 403
21         }
22
23         return $next($request);
24     }
25 }

```

*Ilustración 27: Ejemplo de middleware en Laravel*

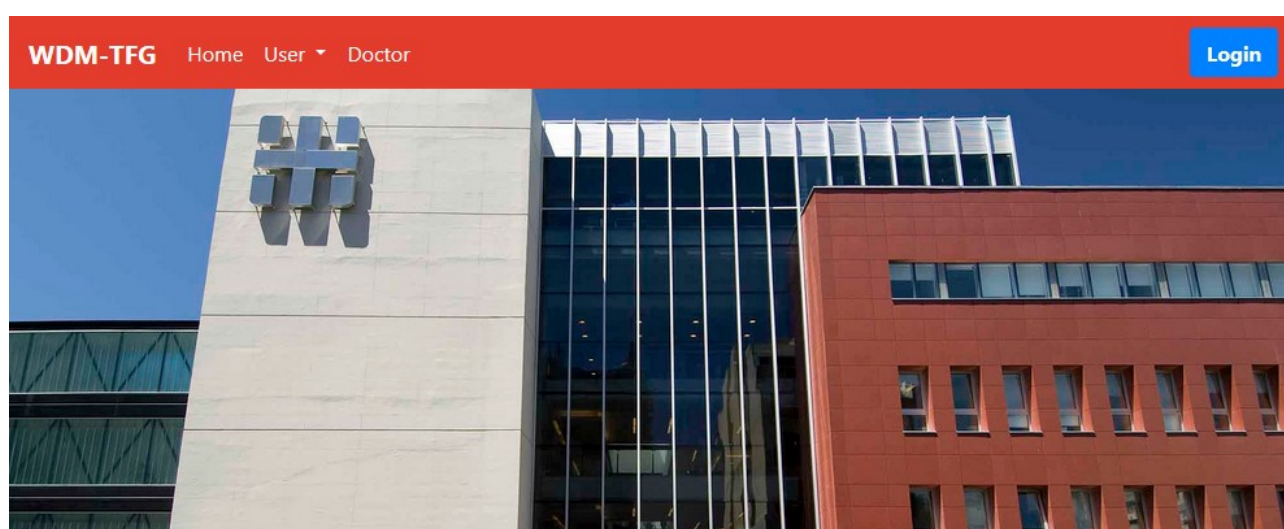
Otra de las funcionalidades muy útiles para la organización de rutas y gestión de acceso son los middleware<sup>[40]</sup>. Consisten en unos archivos o controladores que ejecutan cuando se realiza alguna petición al servidor. Esto permite al usuario definir distintas reglas de cómo debe tratar el sistema el acceso a según qué sitios, por ejemplo. En primer lugar ya aportan un grado de seguridad ya que el hecho de que una ruta deba pasar por un middleware, como en la ilustración 27, indica que el acceso a esa ruta tendrá unas condiciones especiales. En el ejemplo, el usuario debe haber iniciado sesión y, en el middleware, es posible ver cómo los accesos al panel de usuario serán permitidos siempre y cuando el usuario de la web no sea un doctor. En caso de que sea doctor, se devuelve un mensaje de error.

Las funcionalidades de un middleware son muy extensas, pero en el proyecto presente sirve de ejemplo la ilustración mostrada ya que, como se puede comprobar al probar la plataforma, realiza su función de una manera elegante y clara.

## 7. Estructura de la plataforma web

Entrando en la explicación de la plataforma en sí, el alumno ha seguido la propuesta final definida en el apartado 2. Para ello, se ha definido como punto de entrada una web corporativa y se han encerrado las funcionalidades de la página web en diversas áreas privadas.

### 7.1 Web corporativa

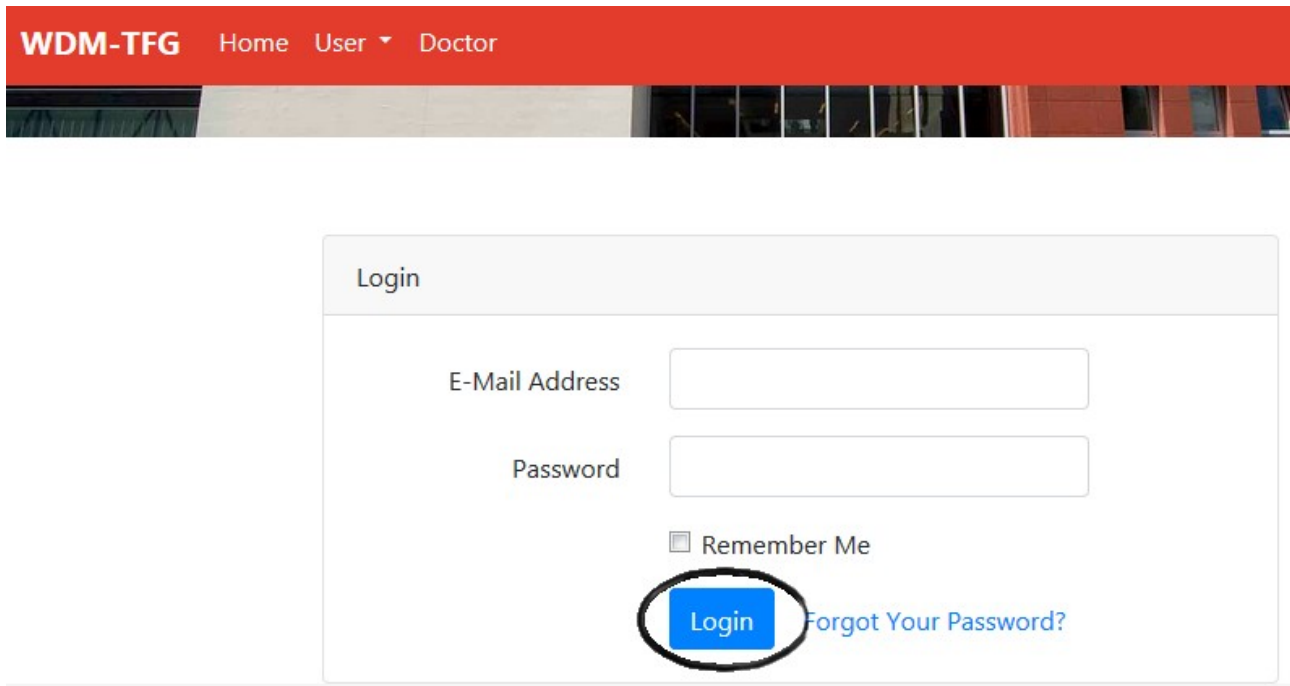


*Ilustración 28: Captura de la página principal del proyecto TFG WDM*

Cuando se realizó el estudio de la idea propuesta se pudo observar que tan solo requería la subida de una imagen para su análisis. Esto se podría haber realizado directamente en una página, pero realizarlo de este modo haría que la web quedase, en primer lugar, poco dinámica y, ante todo, no se aprovechaba las funcionalidades que puede ofrecer una página web. Si nada más entrar a la página web lo que se ve es el formulario para poder subir la imagen se pierde el contacto con el usuario que está accediendo.

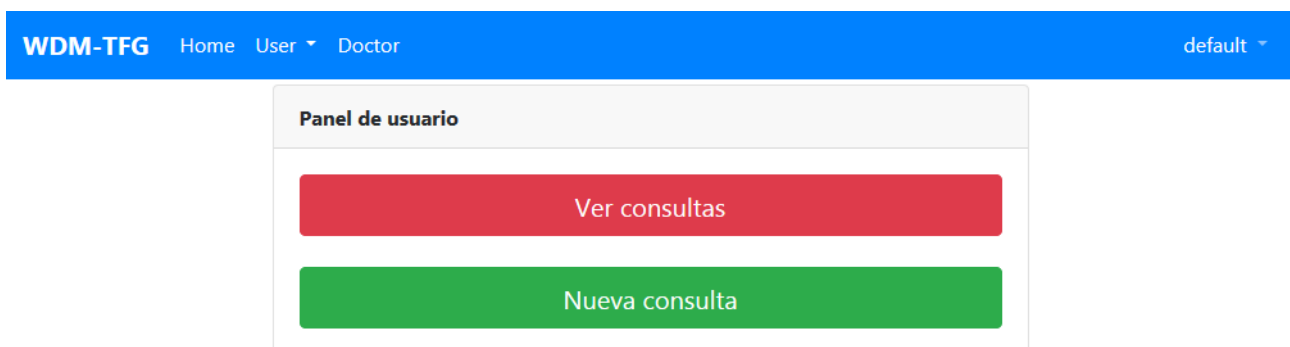
Por el motivo explicado en el párrafo anterior se decidió implementar una página web corporativa, a modo de información. Simplemente el alumno pretende que el punto de entrada a la web sea una página donde pueda encontrar distinta información referente al hospital y al proyecto realizado. Un lugar desde donde el usuario pueda dirigirse a cualquier otra zona de la plataforma.

## 7.2 Área privada



The screenshot shows the top navigation bar of the WDM-TFG web application with links for Home, User, and Doctor. Below the navigation bar is a login form titled "Login". The form contains two input fields: "E-Mail Address" and "Password". Below these fields is a checkbox labeled "Remember Me". At the bottom of the form are two buttons: a blue "Login" button and a blue "Forgot Your Password?" link. The "Login" button is circled in black.

Ilustración 29: Captura del formulario de login



The screenshot shows the user panel of the WDM-TFG web application. The top navigation bar is blue and contains links for Home, User, and Doctor, along with a "default" dropdown menu. Below the navigation bar is a panel titled "Panel de usuario". Inside this panel are two buttons: a red "Ver consultas" button and a green "Nueva consulta" button.

Ilustración 30: Captura del panel de usuario (área privada)

La elección de introducir un área privada para cada una de las partes (admin, usuario y doctor) está planteada para no dejar la página de subida de imágenes en un entorno “público”, al acceso de cualquier tipo de robot o bot que trate de realizar inyección de código HTML o similar.

Además, englobar todas las funcionalidades de cada usuario en cada uno de los paneles privados hace que todo esté más organizado, y que cualquier usuario de un tipo en concreto no pueda acceder a ninguna funcionalidad que no esté indicada para él.

Por último, el alumno pretende utilizar los beneficios de tener una base de datos en el fondo de la aplicación para potenciar el objetivo de la herramienta de análisis dermatológico: pretende introducir un historial de consultas de modo que el usuario pueda tener siempre presente todo lo que ha realizado y la evolución en el tiempo que puede sufrir cada una de sus consultas.

## 8. Desarrollando la aplicación

### 8.1 Comparación entre servidores



*Ilustración 31: Openshift, de Red Hat*



*Ilustración 32: Amazon WebServices (AWS)*

Cuando se está desarrollando una plataforma como la que se explica en este documento, normalmente se realiza bajo un entorno local. El objetivo, sobre todo a nivel de un TFG, siempre es que la aplicación esté operativa para la presentación ante el tribunal. Esta presentación puede realizarse presentando el funcionamiento en el entorno local sin problema, pues enseña claramente que funciona y cómo lo hace.

En este TFG en concreto el desarrollo es para el Hospital Italiano de Buenos Aires. El hospital es quien representa la figura del cliente. Ante este hecho se requiere que para una visualización del funcionamiento de la plataforma, ésta se encuentre en un servidor online que pueda ser accesible vía web. Por lo tanto, no sirve el entorno local.

Quizás lo descrito en el párrafo anterior no supone ningún problema, pues existen muchas alternativas, pero la mayoría de pago. El proyecto actual contempla un escenario de duda y/o incertidumbre sobre una posible adjudicación de presupuesto para contratar un servidor. Es por este motivo que las posibles soluciones que deben encontrarse serían gratuitas.

El tutor de la asignatura, Simone, facilitó la idea usar Openshift Online<sup>[41]</sup>, y ésta era la idea principal a escoger. Las características ofrecidas, aunque quizás menores que una opción de pago, hacen que sea suficiente para el objetivo buscado. En un primer momento la cuenta tardó en ser activada, pues Openshift retrasa la activación de las nuevas cuentas varias semanas por el aumento de usuarios de cuentas gratuitas. Según las consultas realizadas sobre este problema, es algo común y puede demorarse bastante tiempo. Por lo consiguiente, se decide buscar otra alternativa y probar Openshift en otra etapa del proyecto, que se explicará más adelante.

La alternativa escogida es el uso de la capa gratuita de Amazon WebServices (AWS), tal y como se explica en el apartado de tecnología (apartado 5). Esta solución sí que permite poder alojar la plataforma y disponerla online vía web. Esta opción presenta diversos problemas, todos relacionados con el hecho de que la suscripción gratuita en esta plataforma ofrece los servicios de

manera limitada (a nivel de capacidad o de horas de uso). Por el momento, teniendo en cuenta la dimensión de proyecto que se está realizando no se incumpliría ninguno de los límites. Pero el principal problema es que muchos de los servicios que ofrece la suscripción gratuita a AWS tienen una caducidad de un año. Uno de estos servicios es el usado para alojar y mantener online la aplicación. Este problema será importante que sea tenido en cuenta por parte del cliente para un futuro uso de la plataforma.

### 8.1.1 Pruebas en los servidores

En esta sección se explica las pruebas que se han hecho en los servidores gratuitos alternativos y sus resultados. Estos servidores son Hostinger<sup>[42]</sup> (en su plan gratuito) y Openshift Online (en su plan gratuito):

- **Hostinger:** El servidor gratuito de Hostinger es, de todos los servidores probados, el más “básico”. Cumple con las características típicas de un hosting de los más usuales de hace unos años, en el cual el código de la aplicación debe subirse vía FTP (no puede desplegarse desde un repositorio de Github). Este método es menos práctico, ya que la subida se demora bastante (en las pruebas, sobre 30 minutos) y, obligando a volver a pasar los archivos si se producen cambios. Además, al tener que realizar la subida vía FTP se pierde uno de los aspectos positivos explicados de Laravel, y es el uso del gestor de dependencias Composer. Al no poder usar Composer, todos los componentes y librerías que usa el proyecto deben subirse (siendo esta acción la más costosa), a diferencia del repositorio de Github, donde no se suben (al hacer un clone del repositorio, luego mediante el comando de “composer install” ya se descargan todas las dependencias).

Con el código está subido, se puede configurar la BBDD MySQL con PHPMyAdmin mediante otra opción de la plataforma. Es interesante que permita la opción de crear dos BBDD, pues para el proyecto serán necesarias las dos (una para el entorno de producción y otra para los tests).

Una vez está todo configurado, teóricamente la aplicación debería funcionar pero no es así, ya que se muestra un error 500. Esto puede deberse al hecho de que en todos los dispositivos en los cuales se ha podido configurar el proyecto (tanto en el entorno local como en el servidor de Amazon), una vez se ha realizado el clone del repositorio se ha tenido que lanzar unos comandos de configuración para que el proyecto funcionase (tal y como se explica en el anexo). Uno de estos comandos es la generación de la “key” del proyecto. Al no tener acceso al servidor de Hostinger mediante terminal SSH, no se ha podido realizar este paso. Esta key se enlaza al archivo .env del proyecto con las variables de entorno. La subida del archivo .env no es aconsejable (en el repositorio de Github no está por seguridad) ya que contiene información sensible, y para disponer de él en el servidor que se está probando se ha tenido que subir directamente el que había en local, sin éxito.

Por lo consiguiente, se ha descartado el uso de este servidor.

- **Openshift Online:** Este servidor es un poco más avanzado con el anterior, pues tiene un gestor para poder iniciar los distintos tipos de proyecto que se requiera (con distintas tecnologías). A la hora de configurar un proyecto de Laravel mediante el gestor, permite configurar el archivo .env comentado en la prueba anterior. Es cierto que es de gran ayuda, pero siempre es mejor poder crear directamente el archivo de modo tradicional (mediante un editor de texto), ya que ofrece más libertad y seguridad. Además, este servidor sí que permite el despliegue del proyecto mediante el repositorio de Github.

Una vez se han seguido las instrucciones del gestor de despliegue, se presenta el primer problema: solo permite generar una BBDD, lo que provoca que no se pueda usar una BBDD alternativa para los tests.

Por último, después de configurar todo correctamente, el proyecto no ha funcionado. Viendo los logs de errores indica que el servidor utiliza una versión de PHP inferior a la versión de PHP requerida por el proyecto. La versión de PHP requerida por el proyecto viene determinada por los requerimientos que la versión de Laravel tenga. En este caso la versión de Laravel es la 5.7.16, que requiere una versión de PHP 7.1.\*, mientras que el servidor ofrece la versión 7.0.\*.

Como es una versión del servidor gratuita no permite una configuración de las tecnologías propias del servidor, además de que la versión de Laravel a usar debe escogerse a la hora de crear el proyecto, evitando eso poder intentar rebajar la versión de PHP requerida. Es por este motivo por lo que se descarta el uso de este servidor.

Cabe destacar también que en las pruebas de ambos servidores no se permite la opción de configurar adónde deben apuntar inicialmente las peticiones que se realicen. Esto quiere decir que al acceder a la URL servidor no aparece ningún error ni aparece la página, simplemente aparece un listado con los archivos subidos al servidor. Esto se debe a que en los proyectos de Laravel el archivo que recibe todas las peticiones (index.php) está situado dentro de la carpeta “public”, situada en la raíz del proyecto. En el entorno local y en el servidor de Amazon se ha podido configurar el servidor HTTP para que apunte directamente a este archivo, por lo que al acceder aparece directamente la página.

Por último, el aspecto negativo más importante es el hecho de que no ha habido opción de poder configurar de la manera deseada la tecnología de Python. Esto quiere decir que aunque las páginas hubiesen funcionado en las pruebas con los servidores arriba comentados, la ejecución de un script Python no hubiese funcionado (o no correctamente). Quizás en Openshift se da la posibilidad de desplegar un proyecto de Python, pero la elección en el caso que se está explicando es la de desplegar un proyecto Laravel (pues es la tecnología principal). Aún así, aunque pudiese llegar a funcionar Python de manera básica, hay que destacar que la configuración del entorno local y del servidor de Amazon de Python contiene librerías y componentes externos de esa tecnología para poder ejecutar el script de prueba y, con gran seguridad, el futuro script de análisis de lunares.



En el entorno local y en AWS, como se dispone de un acceso total al servidor mediante terminal SSH, es posible configurar todo como si se tuviese físicamente una máquina con ubuntu. Este aspecto descartaría totalmente cualquier servidor en el cual no se pueda configurar con libertad Python conjuntamente con Laravel.

### 8.1.2 Tabla de servidores

A continuación se presenta una tabla con la comparativa de características:

	VCPU	Memoria	Número de BBDD	Número de Proyectos	Despliegue desde GitHub	Acceso SSH	Nivel configurable
<b>Openshift</b>	2	1GiB	1	1	Sí	No	Bajo
<b>Hostinger</b>	1	1GiB	2	2	No	No	Bajo
<b>AWS</b>	1	1GiB	Sin limite	Sin limite	Sí	Sí	Alto

A continuación se presenta una tabla con la comparativa a nivel económico:

	Precio	Requiere datos bancarios	Tiempo de espera de alta
<b>Openshift</b>	Gratis	No	1 mes aprox.
<b>Hostinger</b>	Gratis	No	Inmediato
<b>AWS</b>	Gratis*	Tarjeta (para verificación)	Inmediato

*\*Gratis los primeros 12 meses, 8,50\$ a partir de ese plazo.*

## 8.2 Ejecución de un script Python en Laravel

Uno de los puntos interesantes y más importantes de la aplicación es la posibilidad de ejecutar un script (principalmente de Python<sup>[43]</sup>) en el framework Laravel, que es php. La importancia de esta tarea recae en el hecho de que el algoritmo de análisis dermatológico será un script de Python. El alumno, de este modo, inicia una investigación para ver las posibilidades de ejecutar scripts de Python en Laravel de una manera adecuada.

```
$process = new Process("C:\Users\Victor\Anaconda2\pkgs\python-2.7.15-he216670_0\python V:\Documents\prueba.py argumento1");
$process->run();
$result = $process->getOutput();
```

*Ilustración 33: Ejecución de un script Python en Laravel*

```
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Process\Process;
```

*Ilustración 34: Include de componentes de Symfony*

La solución encontrada consiste en el uso del componente Process<sup>[44]</sup> de Symfony. Tal y como se ha explicado en el apartado 5 de tecnología, el framework Laravel implementa muchos componentes de Symfony. En la ilustración 34 es posible observar cómo el controlador con la lógica de ejecución del script realiza “includes” de Symfony. Además, en la ilustración 33 se puede ver la implementación de la solución encontrada por el alumno.

El resultado de este script es enviado como parámetro a la vista de la página para su uso.

## 8.3 Test Driven Development (TDD)

Test Driven Development (TDD)<sup>[45]</sup> es una metodología de desarrollo de software que consiste en la realización, en primer lugar, de las pruebas que las distintas funcionalidades de una sección deberían pasar para considerar que dichas funcionalidades están implementadas correctamente.

En primer lugar se empieza desarrollando un caso de prueba básico que se tenga previsto que falle, de modo que a medida que se va consiguiendo el resultado esperado en la prueba se va generando el código de la funcionalidad en sí. Esta metodología “contradice” al flujo lógico o normal, en el cual se desarrolla la funcionalidad en sí para acabar realizando la prueba.

Realizando el desarrollo con TDD, se consigue vigilar más cómo se realiza una funcionalidad, previniendo posibles despistes en cuanto a los resultados obtenidos.

La realización de la plataforma web de este TFG se ha realizado, en su mayoría, con TDD (contiene aproximadamente unos 100 tests).

## 9. Posibles ampliaciones

Uno de los requisitos no funcionales que debía tener el proyecto a la hora de su planteamiento era que debía ser escalable para poder introducir mejoras en un futuro. A continuación se detallan aspectos que serían muy interesantes poder introducir.

- **Script Python de análisis de lunar:** La mejora más importante que se podría introducir es el script Python que se encargará de analizar la imagen con el lunar que el usuario suba a la plataforma. Es el aspecto fundamental sobre el que gira el proyecto, ya que se basa en poder proporcionar una estructura web interesante alrededor del script, tal y como se ha realizado, dejando todos los aspectos necesarios configurados y un script funcionando.

Actualmente el script que se ejecuta al subir una imagen es un script de ejemplo ya que, como se ha comentado a lo largo del documento, el desarrollo del algoritmo de análisis no entra dentro de las competencias de este TFG. Es por este motivo que el desarrollo e implementación de dicho algoritmo en la plataforma web es la ampliación más importante, aprovechando que el estado actual de la plataforma permite que el cambio del script de ejemplo por el script futuro sea simplemente cambiar una línea de código.

- **Desarrollo de una app móvil nativa:** Tal y como se ha explicado, la idea inicial de este proyecto siempre ha sido el desarrollo de una plataforma web. A lo largo de las distintas reuniones que se han mantenido por la parte de la UB (tanto Simone Balocco como el alumno) con la parte del Hospital Italiano (José Massa) así se ha comunicado siempre. Pero con la reunión con fecha de 23 de noviembre de 2018 se determinó que se requeriría el desarrollo de una app móvil. Para aprovechar todo el estudio y el trabajo que hasta ese momento se había realizado, se ofreció la opción de utilizar el componente de Android WebView, basado en la visualización de la página web directamente en la pantalla del dispositivo.

Esta solución ofrece la posibilidad de desarrollar una app móvil útil sin tener un impacto muy significativo en la planificación del proyecto. El aspecto negativo es que las características y puntos positivos que puede tener una app desarrollada de manera nativa no serán visibles. Es por este motivo que la realización de una app nativa en Android y/o iOS, realizada exclusivamente como un proyecto en sí, puede aportar al proyecto en general un mayor ámbito de uso, así como una mayor accesibilidad a las características de la aplicación cuando se acceda desde un dispositivo móvil.

- **Modelo 3D del cuerpo:** En las reuniones periódicas con José Massa, intermediario del Hospital Italiano de Buenos Aires, aportó una posible ampliación al proyecto introduciendo un modelo 3D del cuerpo humano. Esta implementación haría que las consultas fuesen más claras, ya que ayudaría a la hora de indicar la zona del cuerpo donde está el lunar de la imagen que se sube.

- **Conexión con servidor de imágenes médicas:** Actualmente toda la información del usuario y de la consulta que se realiza se guarda en una base de datos. José Massa aportó otra posible ampliación dando la posibilidad de que toda la información al respecto del usuario y de la consulta se pudiese incorporar a un fichero de servidor de imágenes médicas (DICOM<sup>[46]</sup>). Almacenando los datos de este modo, se aporta a la imagen no solo la propia imagen en sí, si no un conjunto de datos relacionados (por ejemplo, datos del paciente), haciendo que la imagen quede en un estándar adecuado para su movilidad y almacenamiento.

## 10. Conclusiones

La realización del Trabajo de Fin de Grado “WDM” (Web de dermatología) ha permitido conseguir varias cosas en un plazo corto de tiempo.

El haber realizado este proyecto colaborando con el Hospital Italiano de Buenos Aires ha hecho que el alumno pueda sentir las sensaciones propias de trabajar en un entorno lo más próximo posible con el entorno real, tratando con un cliente (el hospital) que expresa sus requerimientos y va ofreciendo un feedback a medida que avanza el proyecto.

A lo largo del trabajo han habido momentos en los que aparecía algún requerimiento o funcionalidad nueva, teniendo que realizar un nuevo análisis y planificación integrándolos en el desarrollo que se estaba llevando a cabo. Estas situaciones suelen darse en entorno real, y es que los plazos o metodologías que se enseñan en clase tienden a ir casi todas sobre su lado más positivo, prácticamente sin tener en cuenta posibles problemas. Es por este motivo que ha sido interesante poder ver este punto de vista en un proyecto.

Otro aspecto muy gratificante ha sido la posibilidad de poder trabajar con un framework PHP (en este caso). El alumno tenía conocimientos y experiencias previos pero situados varios años atrás, cuando el uso de los frameworks no estaba del todo extendido (o no era tan usual). El poder ver la diferencia de cómo se generaban antes las plataformas en comparación con las facilidades que ofrece un framework ha sido un punto curioso y bonito a la vez.

Por otro lado, aunque el TFG es un trabajo normalmente individual, llevado por el alumno con la supervisión del profesor (Simone Balocco), también ha ofrecido la posibilidad de poder colaborar con otro compañero de la UB (tutelado también por Simone) para intercambiar la información entre los dos a modo de ayuda y de guía. Ha sido gratificante poder ver que alguna solución que se ha implementado en este proyecto tras alguna investigación ha servido también para que el otro compañero pudiese solucionar algún problema suyo. Del mismo modo, observar la ayuda que otro compañero puede ofrecer para desatascar algún problema propio.

Es por estos motivos que el alumno quisiera agradecer al compañero de la UB José Manuel Cano por la colaboración mutua que ha habido durante el proyecto en caso de haberse necesitado.

También agradecer a la persona intermediaria entre el hospital y el alumno (Profesor José Massa) por su dedicación a lo largo del proyecto, siempre procurando obtener la información necesaria del equipo de doctores para que el alumno pudiese avanzar en el trabajo.

Y, por último, al profesor Simone Balocco, por haber ofrecido la idea de realizar esta temática de TFG, así como su ayuda y facilidad para poder realizar el seguimiento con total libertad.

## Bibliografía

- [1] “Oferta Trabajo Final de Grado de la UB, Web de dermatología.” [Online] Disponible: <http://tfg-informatica-ub.wikidot.com/c28f1bb712446d2f4d4988aceb69d874:347> [Acceso: 17/12/2018]
- [2] “Universitat de Barcelona (UB).” [Online] Disponible: <https://www.ub.edu/web/ub/es/index.html>? [Acceso: 17/12/2018]
- [3] “Facultat de Matemàtiques i Informàtica, UB.” [Online] Disponible: <https://mat.ub.edu/> [Acceso: 17/12/2018]
- [4] “Hospital Italiano de Buenos Aires, Argentina.” [Online] Disponible: <https://www.hospitalitaliano.org.ar/> [Acceso: 17/12/2018]
- [5] “Diagnóstico precoz de las enfermedades. Sanitas” [Online] Disponible: <https://www.sanitas.es/sanitas/seguros/es/particulares/biblioteca-de-salud/prevencion-salud/san041825wr.html> [Acceso: 18/12/2018]
- [6] “Un chequeo médico puede detectar una enfermedad potencialmente grave a tiempo. ABC” [Online] Disponible: [https://www.abc.es/salud/enfermedades/abci-chequeo-medico-puede-detectar-enfermedad-potencialmente-grave-tiempo-201606161759\\_noticia.html](https://www.abc.es/salud/enfermedades/abci-chequeo-medico-puede-detectar-enfermedad-potencialmente-grave-tiempo-201606161759_noticia.html) [Acceso: 18/12/2018]
- [7] “La Inteligencia Artificial ya diagnostica enfermedades tan bien como los médicos. La Vanguardia” [Online] Disponible: <https://www.lavanguardia.com/ciencia/cuerpo-humano/20180223/44950677766/inteligencia-artificial-machine-learning-diagnosticar-enfermedades-medicos-eficiencia.html> [Acceso: 18/12/2018]
- [8] “Un nuevo aliado de los médicos. El Mundo” [Online] Disponible: <http://lab.elmundo.es/inteligencia-artificial/salud.html> [Acceso: 18/12/2018]
- [9] “¿Por qué la inteligencia artificial es imprescindible en medicina” [Online] Disponible: <http://www.computing.es/analytics/noticias/1107461046201/inteligencia-artificial-imprescindible-medicina.1.html> [Acceso: 18/12/2018]
- [10] “UNICEN” [Online] Disponible: <https://www.unicen.edu.ar/> [Acceso: 18/12/2018]
- [11] “¿Qué es el diseño web responsive?” [Online] Disponible: <https://www.hosteurope.es/blog/disenio-web-responsivo> [Acceso: 19/12/2018]
- [12] “WebView component, Android” [Online] Disponible: <https://developer.android.com/reference/android/webkit/WebView> [Acceso: 19/12/2018]

- [13] “PHPMyadmin” [Online] Disponible: <https://www.phpmyadmin.net/> [Acceso: 19/12/2018]
- [14] “WampServer” [Online] Disponible: <http://www.wampserver.com/en/> [Acceso: 20/12/2018]
- [15] “WAMP” [Online] Disponible: <https://www.redeszone.net/windows/como-instalar-wamp-un-servidor-http-php-y-mysql-todo-en-uno/> [Acceso: 20/12/2018]
- [16] “Amazon WebServices (AWS)” [Online] Disponible: <https://aws.amazon.com/es/> [Acceso: 20/12/2018]
- [17] “EC2 (AWS)” [Online] Disponible: <https://aws.amazon.com/es/ec2/> [Acceso: 20/12/2018]
- [18] “¿Qué es SSH y cómo funciona?” [Online] Disponible: <https://www.hostinger.es/tutoriales/que-es-ssh> [Acceso: 20/12/2018]
- [19] “Ubuntu” [Online] Disponible: <https://www.ubuntu.com/> [Acceso: 20/12/2018]
- [20] “Apache” [Online] Disponible: <https://www.apache.org/> [Acceso: 20/12/2018]
- [21] “PHP” [Online] Disponible: <http://php.net/manual/es/intro-what-is.php> [Acceso: 20/12/2018]
- [22] “Composer” [Online] Disponible: <https://getcomposer.org/> [Acceso: 20/12/2018]
- [23] “Laravel” [Online] Disponible: <https://laravel.com/> [Acceso: 20/12/2018]
- [24] “Symfony” [Online] Disponible: <https://symfony.com/> [Acceso: 20/12/2018]
- [25] “Laravel Documentation” [Online] Disponible: <https://laravel.com/docs/5.7> [Acceso: 20/12/2018]
- [26] “HTML” [Online] Disponible: <https://developer.mozilla.org/es/docs/Web/HTML> [Acceso: 20/12/2018]
- [27] “CSS” [Online] Disponible: [https://developer.mozilla.org/es/docs/Learn/CSS/Introduction\\_to\\_CSS/Como\\_funciona\\_CSS](https://developer.mozilla.org/es/docs/Learn/CSS/Introduction_to_CSS/Como_funciona_CSS) [Acceso: 20/12/2018]
- [28] “JavaScript” [Online] Disponible: <https://www.javascript.com/> [Acceso: 20/12/2018]
- [29] “Bootstrap” [Online] Disponible: <https://getbootstrap.com/> [Acceso: 20/12/2018]
- [30] “Bootstrap Documentation” [Online] Disponible: <https://getbootstrap.com/docs/4.1/getting-started/introduction/> [Acceso: 20/12/2018]
- [31] “Git” [Online] Disponible: <https://git-scm.com/> [Acceso: 20/12/2018]
- [32] “GitHub” [Online] Disponible: <https://github.com/> [Acceso: 20/12/2018]

- [33] “Visual Studio Code” [Online] Disponible: <https://code.visualstudio.com/> [Acceso: 20/12/2018]
- [34] “Microsoft” [Online] Disponible: <https://www.microsoft.com/es-es/> [Acceso: 20/12/2018]
- [35] “Trello” [Online] Disponible: <https://trello.com/> [Acceso: 20/12/2018]
- [36] “Angular” [Online] Disponible: <https://angular.io/> [Acceso: 20/12/2018]
- [37] “Blade Templates” [Online] Disponible: <https://laravel.com/docs/5.7/blade> [Acceso: 20/12/2018]
- [38] “Eloquent ORM” [Online] Disponible: <https://laravel.com/docs/5.7/eloquent> [Acceso: 20/12/2018]
- [39] “Routing en Laravel” [Online] Disponible: <https://laravel.com/docs/5.7/routing> [Acceso: 20/12/2018]
- [40] “Middleware en Laravel” [Online] Disponible: <https://laravel.com/docs/5.7/middleware> [Acceso: 20/12/2018]
- [41] “Openshift online” [Online] Disponible: <https://www.openshift.com/products/online/> [Acceso: 20/12/2018]
- [42] “Hostinger, plan gratuito” [Online] Disponible: <https://www.hostinger.es/hosting-gratuito> [Acceso: 20/12/2018]
- [43] “Python” [Online] Disponible: <https://www.python.org/> [Acceso: 20/12/2018]
- [44] “Process Component, Symfony” [Online] Disponible: <https://symfony.com/doc/current/components/process.html> [Acceso: 20/12/2018]
- [45] “Test Driven Development (TDD), de José Ignacio Herranz” [Online] Disponible: <https://www.paradigmadigital.com/dev/tdd-como-metodologia-de-diseno-de-software/> [Acceso: 20/12/2018]
- [46] “DICOM, de Javier Pascau (Hospital Gregorio Marañón)” [Online, PDF] Disponible: [http://www.conganat.org/seis/normalizacion05/taller\\_2\\_organizaciones/dicom.pdf](http://www.conganat.org/seis/normalizacion05/taller_2_organizaciones/dicom.pdf) [Acceso: 20/12/2018]
- [47] “GitHub garciashorty (alumno), repositorio del proyecto” [Online] Disponible: [https://github.com/garciashorty/WDM\\_TFG\\_UB](https://github.com/garciashorty/WDM_TFG_UB) [Acceso: 20/12/2018]
- [48] “Composer download” [Online] Disponible: <https://getcomposer.org/download/> [Acceso: 20/12/2018]
- [49] “WDM TFG” [Online] Disponible: <http://35.178.235.252> [Acceso: 20/12/2018]



## 14. Anexo

### 14.1 Configuración del entorno

A continuación se detalla cómo se ha realizado la configuración del entorno del servidor online en el cual se aloja la plataforma web. El repositorio del proyecto puede encontrarse en la página de GitHub del alumno<sup>[47]</sup>.

#### 14.1.1 Servidor

El servidor usado es Amazon WebServices (AWS). Es posible crear una cuenta en la url de la referencia [16]. Una vez se ha creado una cuenta y se ha iniciado sesión, el usuario debe dirigirse a la consola de administración.



*Ilustración 35: Opción EC2 en la consola de administración de AWS*

Una vez que el usuario se encuentra en la consola de administración, debe acceder a la opción “EC2” (ilustración 35) que permitirá crear una máquina virtual online.

#### Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

**Launch Instance**

Note: Your instances will launch in the EU West (London) region

*Ilustración 36: Creando una instancia de máquina virtual*

Tal y como se ve en la ilustración 36, para crear una instancia debe pulsarse el botón Launch Instance.



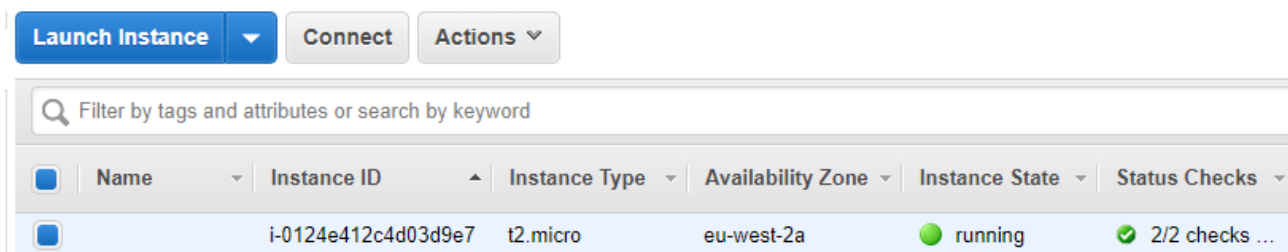
*Ilustración 37: Imagen del SO a elegir para la instancia*

Acto seguido, AWS pregunta por el tipo de sistema operativo a elegir. En la ilustración 37 se muestra que el tipo debe ser Ubuntu Server 18.04 LTS.



*Ilustración 38: Imagen del botón Review and Launch*

En la siguiente página no hay que modificar ninguna opción, simplemente pulsar en el botón azul “Review and Launch”, visible en la ilustración 38, para finalizar la creación de instancia. En caso de que la aplicación pregunte por una “key”, se debe crear una nueva con el nombre deseado y guardarla en el lugar del PC que el usuario quiera.



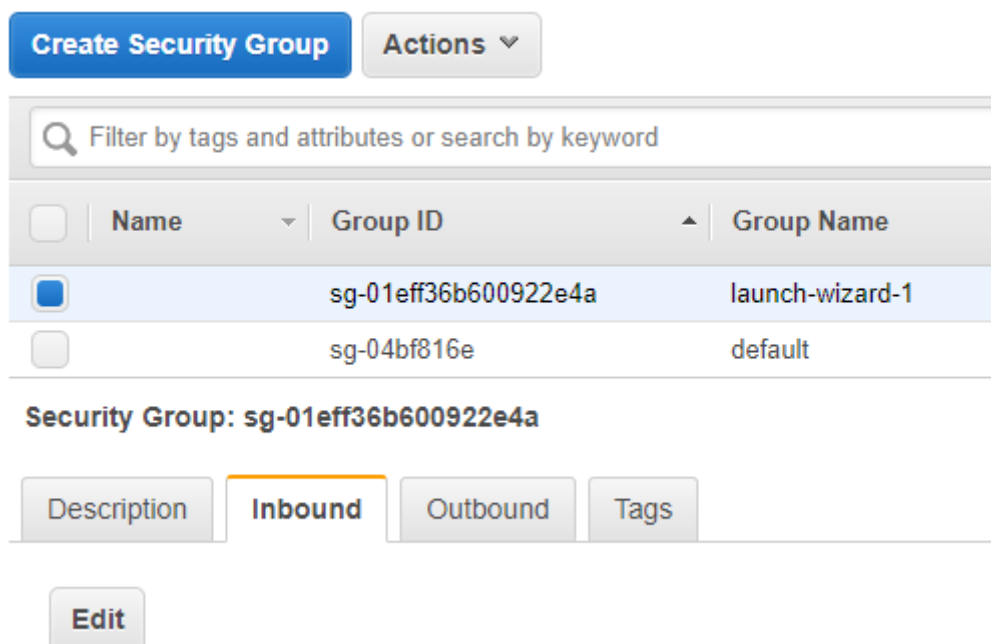
*Ilustración 39: Captura de la instancia creada*



*Ilustración 40:  
Muestra del menú de  
redes y seguridad*

En la pantalla de la ilustración 39 se puede ver la instancia creada.

La siguiente acción a realizar es dirigirse a la opción “Security Groups” visible en la ilustración 40.



*Ilustración 41: Captura de la pantalla de "Security Groups"*

En la pantalla de “Security Groups”, debe marcarse la primera opción (ilustración 41) y, en la pestaña “Inbound”, pulsar el botón “Edit”.

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
HTTP ▼	TCP	80	Custom ▼ 0.0.0.0/0
HTTP ▼	TCP	80	Custom ▼ ::/0
SSH ▼	TCP	22	Custom ▼ 0.0.0.0/0

Add Rule

*Ilustración 42: Creando una regla HTTP para la instancia*

En el marco que se abre al pulsar “Edit”, el usuario debe añadir, mediante el botón “Add Rule”, una regla de tipo HTTP como se muestra en la ilustración 42.

Con estas acciones, el usuario ya tendría lista la instancia de máquina virtual de EC2 online.

### 14.1.2 Servidor HTTP Apache

```
$ ssh -i wdm_tfg.pem ubuntu@35.178.235.252
```

*Ilustración 43: Comando para la conexión por SSH a la instancia creada*

Para realizar el resto de configuraciones es necesario conectarse a la instancia creada. Para ello se usará SSH (conexión mediante línea de comandos). Con el comando de la ilustración 43 es posible realizar la conexión (el parámetro “wdm\_tfg.pem” es la key que se creó en el apartado anterior, y ubuntu@IP son las credenciales y dirección IP).

```
ubuntu@ip-172-31-26-13:~$ sudo apt-get update
```

*Ilustración 44: Comando apt-get update*

```
ubuntu@ip-172-31-26-13:~$ sudo apt-get upgrade
```

*Ilustración 45: Comando apt-get upgrade*

Al conectarse a la instancia, como está recién creada, es importante realizar los comandos indicados en las ilustraciones 44 y 45 con tal de refrescar los repositorios.

```
ubuntu@ip-172-31-26-13:~$ sudo apt-get install apache2|
```

*Ilustración 46: Comando para instalar apache2*

Acto seguido ya es posible empezar a instalar servicios. El primero será Apache2, mediante el comando de la ilustración 46.

```
ubuntu@ip-172-31-26-13:~$ sudo a2enmod rewrite|
```

*Ilustración 47: Activando mod rewrite de apache2*

Es conveniente activar el módulo de Apache que viene instalado por defecto (pero quizás no activo) mod\_rewrite. Este módulo sirve para crear URL alternativas, en lugar de las generadas automáticamente (ilustración 47).

```
ubuntu@ip-172-31-26-13:~$ sudo nano /etc/apache2/apache2.conf
ubuntu@ip-172-31-26-13:~$ |
```

*Ilustración 48: Editar el archivo de configuración de Apache*

Se debe modificar el archivo de configuración de Apache para adecuarlo al proyecto. Con el comando de la ilustración 48 se puede acceder al editor.

```
# These need to be set in /etc/apache2/envvars
#User ${APACHE_RUN_USER}
#Group ${APACHE_RUN_GROUP}
User ubuntu
Group ubuntu
```

*Ilustración 49: Configurando Apache (1)*

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory /home/ubuntu/ >
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

*Ilustración 50: Configurando Apache (2)*

En el archivo de configuración que se abre, se debe modificar las líneas de la ilustración 49, comentando las primeras instancias de “user” y “group” para establecer las que están activas en la ilustración.

Por otro lado, un poco más abajo, es posible encontrar el código de la ilustración 50. Lo que se debe hacer en este caso es copiar la primera directiva “Directory”, y crear una justo debajo, con los datos que se ve en la ilustración.

Una vez se han realizado los cambios, con ctrl+x se puede guardar y salir.

```
ubuntu@ip-172-31-26-13:~$ sudo nano /etc/apache2/sites-enabled/000-default.conf
```

*Ilustración 51: Accediendo al archivo de configuración de sitios activos*

A continuación se debe configurar los sitios activos. Se puede acceder mediante el comando de la ilustración 51.

```
# However, you must set it for any further v
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /home/ubuntu/WDW_TFG_UB/public
```

*Ilustración 52: Configurando los sitios activos para que apunten al proyecto*

En el archivo que se abre se debe modificar la línea de la ilustración 52 con los datos que se ven en la misma. En estas líneas se configura apache para que su ruta principal apunte hacia la ruta principal del proyecto (más adelante se copiará el proyecto a la instancia, pero la ruta será la que indica en la ilustración). Esto quiere decir que cuando en el navegador se introduzca la IP de la instancia, irá directamente a la carpeta public del proyecto, que contiene el archivo de entrada que redirige todas las peticiones.

### 14.1.3 PHP

```
ubuntu@ip-172-31-26-13:~$ sudo apt-get install php|
```

*Ilustración 53: Comando para instalar PHP*

Para instalar PHP se debe ejecutar el comando mostrado en la ilustración 54.

```
ubuntu@ip-172-31-26-13:~$ sudo apt-get install libapache2-mod-php|
```

*Ilustración 54: Comando de instalación de libapache2-mod-php*

```
ubuntu@ip-172-31-26-13:~$ sudo apt-get install php-xml|
```

*Ilustración 55: Comando de instalación de php-xml*

```
ubuntu@ip-172-31-26-13:~$ sudo apt-get install php-mcrypt|
```

*Ilustración 56: Comando de instalación de php-mcrypt*

Es conveniente también instalar los componentes indicados en las ilustraciones 54, 55 y 56 con los comandos señalados. Estos componentes son libapache2-mod-php, php-xml y php-mcrypt.

### 14.1.4 MySQL

```
ubuntu@ip-172-31-26-13:~$ sudo apt-get install mysql-server|
```

*Ilustración 57: Comando de instalación de mysql-server*

En este paso, se instalará el servidor de base de datos, en concreto, MySQL. El comando de instalación puede verse en la ilustración 57. Es importante estar atento en esta acción, ya que posiblemente se pida establecer una contraseña, que se debe recordar.

## 14.1.5 Composer

### Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === '93b54496392c062774670ac18b134c3b3a95e5a5e5c8f1a9f115
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

*Ilustración 58: Comandos a ejecutar para instalar Composer obtenidos directamente de la página oficial*

Acto seguido se realizará la instalación de composer. Para ello el usuario debe dirigirse a la página oficial, concretamente a la sección de download<sup>[48]</sup>. En dicha sección de download se puede ver la sección “Command-line installation”, que contiene los comandos a ejecutar para la instalación (ilustración 58).

### Globally #

You can place the Composer PHAR anywhere you wish. If you put it i  
On unixy systems you can even make it executable and invoke it with

After running the installer following [the Download page instructions](#) y  
path:

```
mv composer.phar /usr/local/bin/composer
```

*Ilustración 59: Comando para ejecutar Composer desde cualquier lugar*

El comando de la ilustración 59 sirve para poder ejecutar los comandos de composer desde cualquier lugar de la aplicación, de manera global. Es importante ejecutar dicho comando con la opción “sudo” delante.



### 14.1.6 Git

```
ubuntu@ip-172-31-26-13:~$ sudo apt-get install git-core
```

*Ilustración 60: Comando de instalación de Git*

La instalación de Git normalmente no será necesaria, ya que el sistema ya lo trae preinstalado. Si no fuese de esta manera, mediante el comando de la ilustración 60 se podría solucionar.

### 14.1.7 PHPMyAdmin

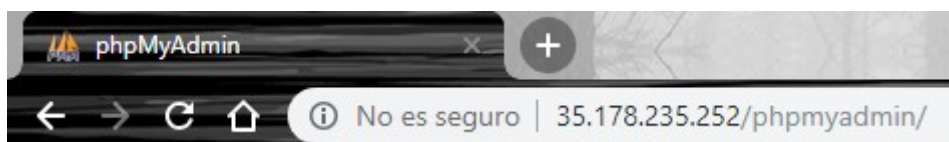
```
ubuntu@ip-172-31-26-13:~/WDM_TFG_UB$ sudo apt-get install phpmyadmin php-mbstring php-gettext
```

*Ilustración 61: Comando de instalación del cliente PHPMyAdmin*

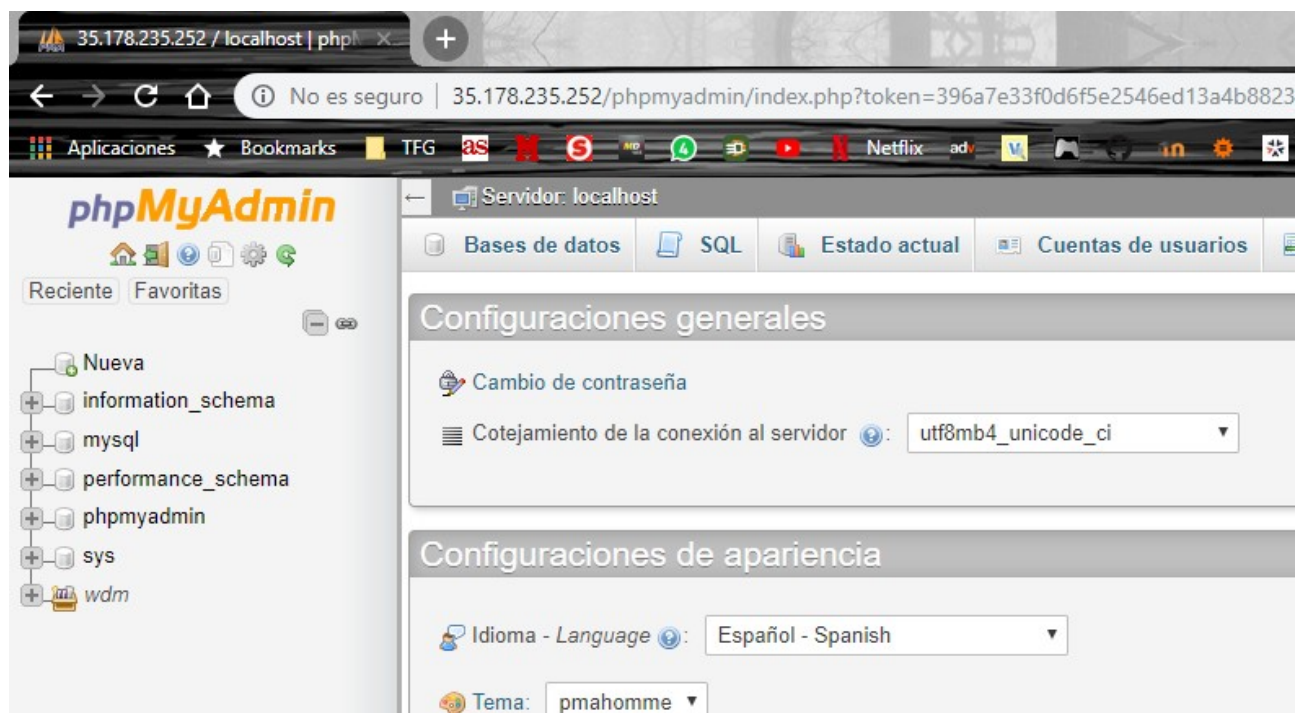
Con el comando de la ilustración 61 se puede instalar el cliente PHPMyAdmin para poder editar la base de datos MySQL del servidor que se ha desplegado en puntos anteriores. El punto positivo de este cliente es que se accede mediante el navegador web.

Al igual que en la instalación del servidor MySQL, es importante prestar atención a la hora de establecer la contraseña que se pedirá en el proceso, pues esta contraseña será la que se deberá utilizar en el proceso de login.

Es aconsejable ejecutar el comando “sudo service apache2 restart” para reiniciar el servidor web y que procese los cambios realizados.



*Ilustración 62: URL de acceso al cliente web de PHPMyAdmin*



*Ilustración 63: Captura de la página principal del cliente de PHPMyAdmin*

Mediante la url mostrada en la ilustración 62 se puede acceder al cliente PHPMyAdmin. Si al acceder se requiere un usuario y contraseña, se puede introducir como usuario “root ó admin” y como contraseña la que se ha establecido a la hora de instalar el cliente.

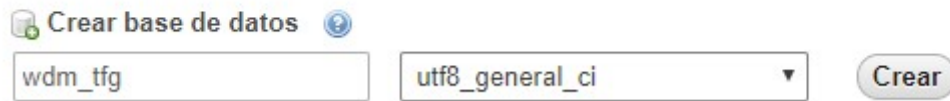
Al entrar en el cliente se podrá ver una página como la mostrada en la ilustración 63. Aunque es un paso opcional, sería importante crear una cuenta de usuario nueva para no usar la de root o admin. Esto es posible en la barra de menú superior, en la opción “Cuentas de usuarios” visible en la captura.



*Ilustración 64: Opción para crear una base de datos*

Mediante la opción “nueva” visible en la ilustración 64 se puede crear una base de datos.

## Bases de datos

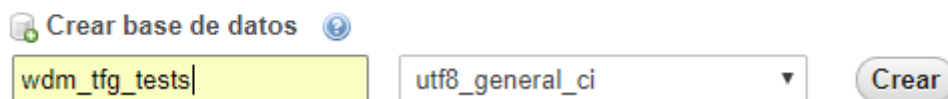


Crear base de datos ?

wdm\_tfg utf8\_general\_ci ▼ Crear

*Ilustración 65: Nombre y tipo para la base de datos*

## Bases de datos



Crear base de datos ?

wdm\_tfg\_tests| utf8\_general\_ci ▼ Crear

*Ilustración 66: Nombre y tipo para la base de datos de tests*

Se deben crear dos bases de datos: la original (la que usará la plataforma, ilustración 65); y la de tests (se usará solamente en los tests, ilustración 66). Esto se realiza de este modo para que al realizar los tests no se modifique accidentalmente la base de datos original.

### 14.1.8 Laravel

```
ubuntu@ip-172-31-26-13:~$ git clone https://github.com/garciashorty/WDM_TFG_UB.git|
```

*Ilustración 67: Comando para realizar un clone del repositorio de Github*

Para desplegar el proyecto que se ha realizado no es necesario instalar ningún proyecto de Laravel, puesto que el proyecto ya está desplegado en el repositorio de Github (en este caso, en el personal del alumno).

De este modo, para poder obtener el proyecto, se ejecutará el comando “clone” de Git. Se puede ver un ejemplo en la ilustración 67.

```
ubuntu@ip-172-31-26-13:~$ ls
WDM_TFG_UB
ubuntu@ip-172-31-26-13:~$ ls WDM_TFG_UB/
app      composer.json  database      public      routes      tests
artisan  composer.lock  package.json  readme.md   server.php   vendor
bootstrap config         phpunit.xml   resources   storage      webpack.mix.js
ubuntu@ip-172-31-26-13:~$ |
```

*Ilustración 68: Comprobación de que se ha clonado el repositorio correctamente*

Al realizar el comando de “git clone”, el repositorio se debe descargar en la máquina virtual (ilustración 68). La siguiente acción se debe realizar dentro del directorio descargado.

```
ubuntu@ip-172-31-26-13:~/WDM_TFG_UB$ composer install|
```

*Ilustración 69: Comando para descargar las dependencias del proyecto con Composer*

Dentro del directorio del proyecto es necesario realizar el comando “composer install” (ilustración 69). Este comando lo que hace es descargar todas las dependencias del proyecto, ya que no se suben al repositorio. No es algo muy importante, ya que esa es la parte positiva del gestor de dependencias (Composer). Mediante el comando indicado se descargarán todas y cada una de ellas.

```

ubuntu@ip-172-31-26-13:~/WDM_TFG_UB$ cat .env
APP_NAME=WDM-TFG
APP_ENV=local
APP_KEY=base64:3D0v7RJ+V4gvr1JhWSCqeD2c700f+6R77wd4W/vfLoY=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=wdm_tfg
DB_USERNAME=root
DB_PASSWORD=WDM_tfg_UB_vgd

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
ubuntu@ip-172-31-26-13:~/WDM_TFG_UB$ |

```

*Ilustración 70: Contenido del archivo .env*

En la raíz del directorio del proyecto se debe crear un archivo llamado “.env” que contendrá las variables de entorno de la plataforma requeridas por Laravel (ilustración 70).

En este caso es importante fijarse en los valores de “DB\_DATABASE” (nombre de la base de datos original), “DB\_USERNAME” (nombre del usuario de la base de datos creado) y “DB\_PASSWORD”(contraseña del usuario creado). Se puede modificar también el valor de “APP\_NAME” para establecer el nombre de nuestra plataforma.

```
<php>
  <env name="APP_ENV" value="testing"/>
  <env name="BCRYPT_ROUNDS" value="4"/>
  <env name="CACHE_DRIVER" value="array"/>
  <env name="MAIL_DRIVER" value="array"/>
  <env name="QUEUE_CONNECTION" value="sync"/>
  <env name="SESSION_DRIVER" value="array"/>
  <env name="DB_DATABASE" value="wdm_tfg_tests"/>
</php>
```

*Ilustración 71: Código del archivo phpunit.xml*

En el archivo “phpunit.xml” de la raíz del directorio del proyecto se debe la última línea “env” de la directiva “php” (ilustración 71). Con esta línea se indica a Laravel que se usará una base de datos en concreto para los tests.

```
ubuntu@ip-172-31-26-13:~/WDM_TFG_UB$ php artisan key:generate
```

*Ilustración 72: Comando de generación de clave*

El comando de la ilustración 72 generará una nueva clave para el archivo .env creado más arriba.

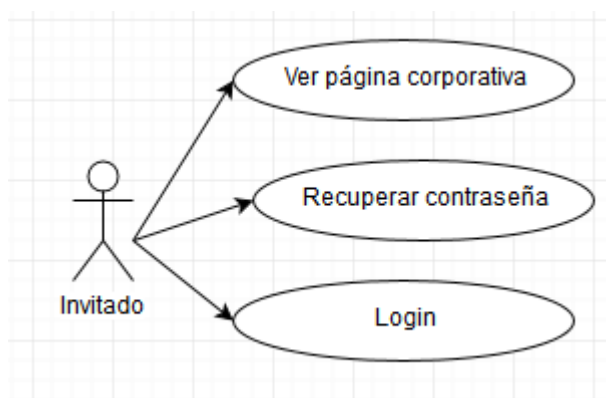
```
ubuntu@ip-172-31-26-13:~/WDM_TFG_UB$ php artisan migrate:fresh --seed
```

*Ilustración 73: Comando para realizar una migración desde 0 de la base de datos*

Por último, se realizará una migración completa de la base de datos del proyecto con los archivos que contiene Laravel (que el desarrollador ha modificado), con el comando de la ilustración 73. Lo que hará el comando será volver a crear todas las tablas (borrando previamente en caso de que haya algún dato) y, además, ejecutará los “seeders” (archivos que sirven para introducir datos para la base de datos al realizar las migraciones).

## 14.2 Casos de uso

### 14.2.1 Usuario invitado



*Ilustración 74: Casos de uso de invitado*

<b>Caso de uso 1.</b> Ver página corporativa	
<b>Objetivo</b>	Ver la información de la página corporativa
<b>Autor</b>	Invitado
<b>Precondiciones:</b> -	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Navega por la página corporativa</li> </ol>	
<b>Extensiones</b>	

<b>Caso de uso 2.</b> Recuperar contraseña	
<b>Objetivo</b>	Recuperar la contraseña de un usuario del sistema mediante el email.
<b>Autor</b>	Invitado
<b>Precondiciones:</b> Disponer de una cuenta de usuario	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Accede a la página de login</li> <li>4. Sistema: Presenta la página de login</li> <li>5. Usuario: Hace click en recuperar contraseña</li> </ol>	

6. Sistema: Presenta la página de recuperación de contraseña
7. Usuario: Indica el email de la cuenta
8. Sistema: Validación del email
9. Sistema: Genera una nueva contraseña al usuario
  - 9.1. Envía email al usuario
10. Sistema: Muestra mensaje de confirmación

**Extensiones**

- 6.1 Usuario no introduce un email correcto. Vuelve al paso 4.
- 6.2 Muestra de mensajes de error en datos introducidos.

**Caso de uso 3. Login**

<b>Objetivo</b>	Usuario inicia sesión en la plataforma.
<b>Autor</b>	Invitado

**Precondiciones:** Disponer de una cuenta de usuario

**Procedimiento**

1. Usuario: Visita la página WDM
2. Sistema: Procesa la petición
3. Usuario: Accede a la página de login
4. Sistema: Presenta la página de login
5. Usuario: Introduce el usuario y la contraseña
6. Sistema: Validación del email
7. Sistema: Autentica al usuario en la plataforma
8. Sistema: Se redirecciona al panel de usuario-doctor

**Extensiones**

- 6.1 Usuario no introduce un email correcto. Vuelve al paso 4.
- 6.2 Muestra de mensajes de error en datos introducidos.



### 14.2.2 Usuario por defecto

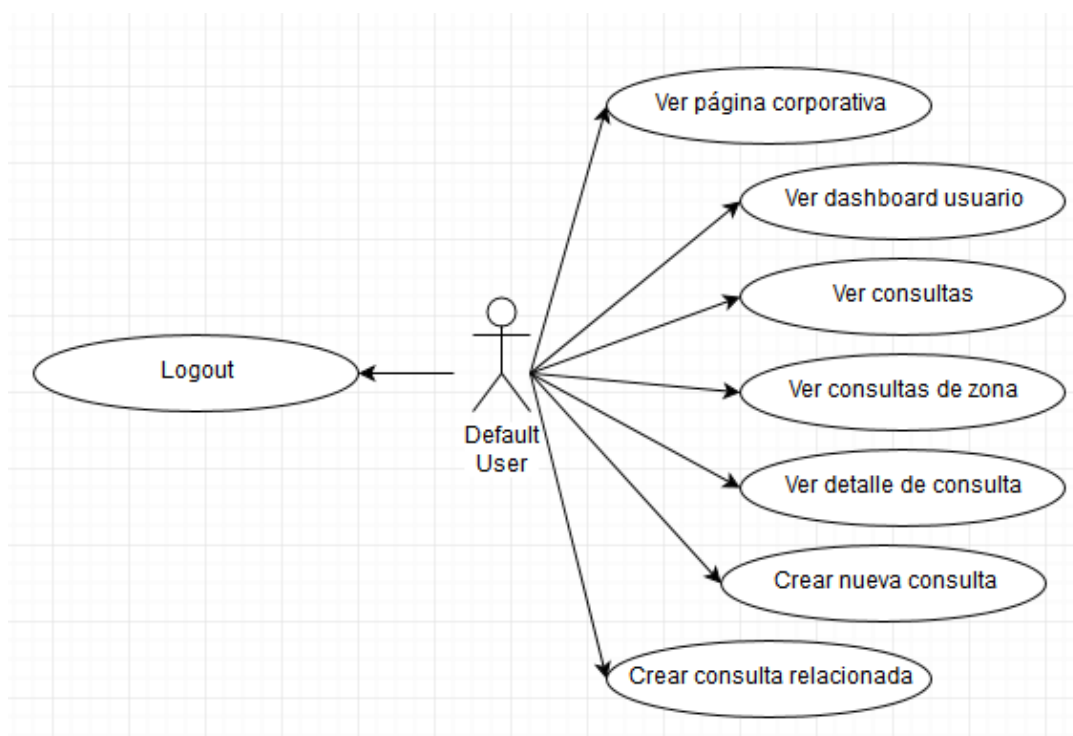


Ilustración 75: Casos de uso de usuario

Caso de uso 4. Logout	
<b>Objetivo</b>	Un usuario autenticado se desconecta del sistema
<b>Autor</b>	Usuario
<b>Precondiciones:</b> El usuario está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Selecciona la opción “logout”</li> <li>4. Sistema: Procesa la petición y desconecta al usuario del sistema</li> <li>5. Sistema: Redirige a la página principal</li> </ol>	
<b>Extensiones</b>	

Caso de uso 5. Ver página corporativa	
<b>Objetivo</b>	Ver la información de la página corporativa
<b>Autor</b>	Usuario
<b>Precondiciones:</b> El usuario está logeado en el sistema	
<b>Procedimiento</b>	

1. Usuario: Visita la página WDM
2. Sistema: Procesa la petición
3. Usuario: Navega por la página corporativa

**Extensiones****Caso de uso 6.** Ver dashboard de usuario

<b>Objetivo</b>	Ver el panel de usuario
<b>Autor</b>	Usuario

**Precondiciones:** El usuario está logeado en el sistema**Procedimiento**

1. Usuario: Visita la página WDM
2. Sistema: Procesa la petición
3. Usuario: Hace click en el enlace “usuario”
4. Sistema: Procesa la petición y presenta la página
5. Usuario: Visualiza el dashboard de usuario

**Extensiones**

3.1 Si el usuario no ha hecho login o si lo ha hecho pero no es un usuario (es doctor o admin), se presenta la página de login.

**Caso de uso 7.** Ver listado de consultas

<b>Objetivo</b>	Ver el listado de consultas que ha realizado
<b>Autor</b>	Usuario

**Precondiciones:** El usuario está logeado en el sistema**Procedimiento**

1. Usuario: Visita la página WDM
2. Sistema: Procesa la petición
3. Usuario: Hace click en el enlace “usuario”
4. Sistema: Procesa la petición y presenta la página
5. Usuario: Selecciona la opción “ver consultas”
6. Sistema: Procesa la petición y presenta la página con el listado de consultas
7. Usuario: Visualiza la lista de consultas realizadas

**Extensiones**

3.1 Si el usuario no ha hecho login o si lo ha hecho pero no es un usuario (es doctor o admin), se presenta la página de login.

**Caso de uso 8.** Ver listado de consultas por zona del cuerpo

<b>Objetivo</b>	Ver el listado de consultas por zona del cuerpo
-----------------	---

<b>Autor</b>	Usuario
<b>Precondiciones:</b> El usuario está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “usuario”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver consultas”</li> <li>6. Sistema: Procesa la petición y presenta la página con el listado de consultas</li> <li>7. Usuario: Selecciona una zona del cuerpo donde se ha realizado una consulta</li> <li>8. Sistema: Procesa la petición y presenta el listado de consultas filtrado por esa zona</li> <li>9. Usuario: Visualiza el listado de consultas filtrado por zona</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el usuario no ha hecho login o si lo ha hecho pero no es un usuario (es doctor o admin), se presenta la página de login.</li> </ol>	

<b>Caso de uso 9.</b> Ver detalle de una consulta	
<b>Objetivo</b>	Ver en detalle la información de una consulta
<b>Autor</b>	Usuario
<b>Precondiciones:</b> El usuario está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “usuario”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver consultas”</li> <li>6. Sistema: Procesa la petición y presenta la página con el listado de consultas</li> <li>7. Usuario: Selecciona una zona del cuerpo donde se ha realizado una consulta</li> <li>8. Sistema: Procesa la petición y presenta el listado de consultas filtrado por esa zona</li> <li>9. Usuario: Hace click en el icono de “ver detalle” de una consulta</li> <li>10. Sistema: Procesa la petición y presenta la página de detalle de una consulta</li> <li>11. Usuario: Visualiza los detalles de una consulta</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el usuario no ha hecho login o si lo ha hecho pero no es un usuario (es doctor o admin), se presenta la página de login.</li> </ol>	

<b>Caso de uso 10.</b> Crear una nueva consulta	
<b>Objetivo</b>	Crear una nueva consulta
<b>Autor</b>	Usuario

<b>Precondiciones:</b> El usuario está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “usuario”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “nueva consulta”</li> <li>6. Sistema: Procesa la petición y presenta el formulario para una nueva consulta</li> <li>7. Usuario: Rellena los campos y envía los datos</li> <li>8. Sistema: Valida los datos</li> <li>9. Sistema: Procesa la petición y crea una consulta</li> <li>10. Sistema: Redirige a la lista de consultas</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el usuario no ha hecho login o si lo ha hecho pero no es un usuario (es doctor o admin), se presenta la página de login.</li> <li>8.1 Si los datos no son válidos, se vuelve al paso 6.</li> <li>8.2 Muestra mensajes de error, en caso de que haya.</li> </ol>	

<b>Caso de uso 11.</b> Crear una consulta relacionada	
<b>Objetivo</b>	Crear una consulta relacionada con una zona ya consultada
<b>Autor</b>	Usuario
<b>Precondiciones:</b> El usuario está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “usuario”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver consultas”</li> <li>6. Sistema: Procesa la petición y presenta la página con el listado de consultas</li> <li>7. Usuario: Selecciona una zona del cuerpo donde se ha realizado una consulta</li> <li>8. Sistema: Procesa la petición y presenta el listado de consultas filtrado por esa zona</li> <li>9. Usuario: Hace click en el icono de “ver detalle” de una consulta</li> <li>10. Sistema: Procesa la petición y presenta la página de detalle de una consulta</li> <li>11. Usuario: Selecciona la opción “Generar consulta relacionada”</li> <li>12. Sistema: Procesa la petición y presenta el formulario para consulta relacionada</li> <li>13. Usuario: Rellena los campos y envía los datos</li> <li>14. Sistema: Valida los datos</li> <li>15. Sistema: Procesa la petición y crea una consulta relacionada</li> <li>16. Sistema: Redirige al listado de consultas</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el usuario no ha hecho login o si lo ha hecho pero no es un usuario (es doctor o admin), se presenta la página de login.</li> <li>14.1 Si los datos no son válidos, se vuelve al paso 6.</li> </ol>	

## 14.2 Muestra mensajes de error, en caso de que haya.

## 14.2.3 Doctor

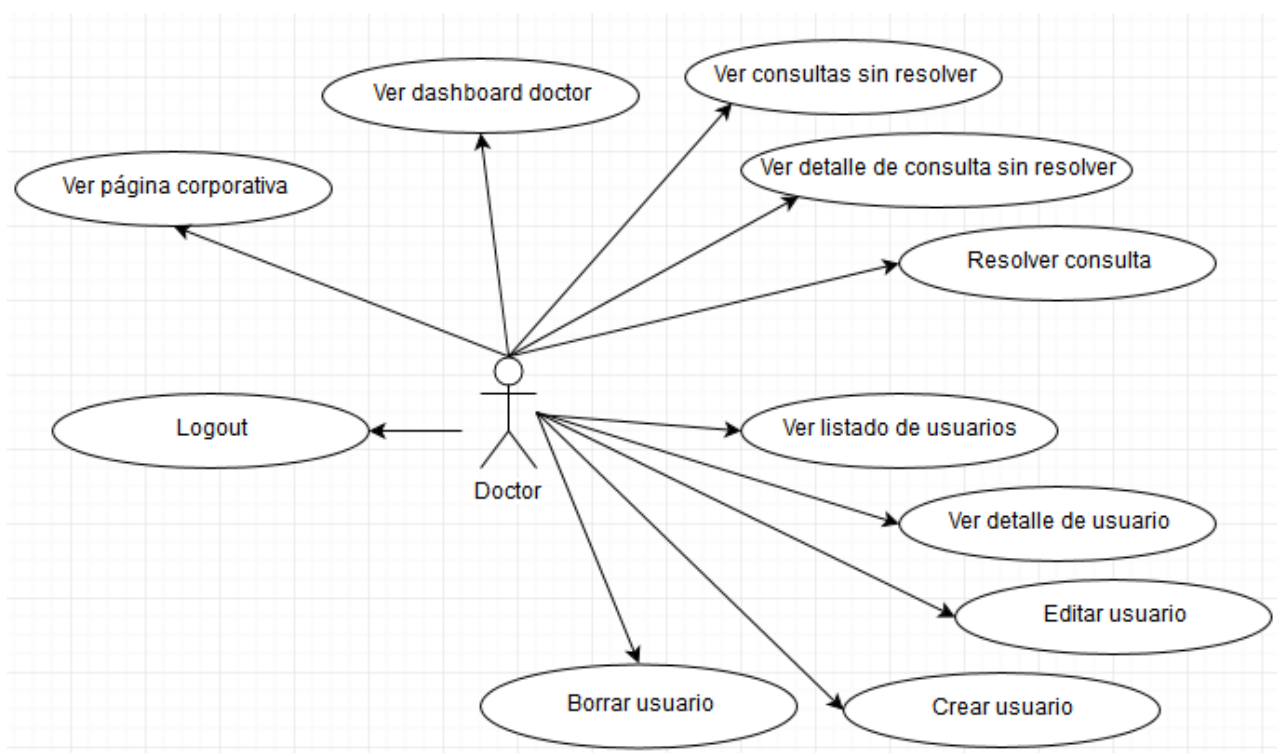


Ilustración 76: Casos de uso de doctor

<b>Caso de uso 12. Logout</b>	
<b>Objetivo</b>	Un doctor autenticado se desconecta del sistema
<b>Autor</b>	Doctor
<b>Precondiciones:</b> El doctor está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>6. Usuario: Visita la página WDM</li> <li>7. Sistema: Procesa la petición</li> <li>8. Usuario: Selecciona la opción “logout”</li> <li>9. Sistema: Procesa la petición y desconecta al doctor del sistema</li> <li>10. Sistema: Redirige a la página principal</li> </ol>	
<b>Extensiones</b>	

<b>Caso de uso 13.</b> Ver listado de consultas sin resolver	
<b>Objetivo</b>	Ver el listado de consultas sin resolver en el sistema
<b>Autor</b>	Doctor
<b>Precondiciones:</b> El doctor está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “doctor”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver consultas no resueltas”</li> <li>6. Sistema: Procesa la petición y presenta la página con el listado de consultas no resueltas</li> <li>7. Usuario: Visualiza la lista de consultas no resueltas</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el doctor no ha hecho login o si lo ha hecho pero no es un usuario (es doctor o admin), se presenta la página de login.</li> </ol>	

<b>Caso de uso 14.</b> Ver detalle de una consulta no resuelta	
<b>Objetivo</b>	Ver en detalle la información de una consulta no resuelta
<b>Autor</b>	Doctor
<b>Precondiciones:</b> El doctor está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “doctor”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver consultas no resueltas”</li> <li>6. Sistema: Procesa la petición y presenta la página con el listado de consultas no resueltas</li> <li>7. Usuario: Hace click en el icono de “ver detalle” de una consulta no resuelta</li> <li>8. Sistema: Procesa la petición y presenta la página de detalle de una consulta no resuelta</li> <li>9. Usuario: Visualiza los detalles de una consulta no resuelta</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el doctor no ha hecho login o si lo ha hecho pero no es un usuario (es doctor o admin), se presenta la página de login.</li> </ol>	

<b>Caso de uso 15.</b> Resolver una consulta	
<b>Objetivo</b>	Resolver una consulta no resuelta previamente
<b>Autor</b>	Doctor
<b>Precondiciones:</b> El doctor está logeado en el sistema	

**Procedimiento**

1. Usuario: Visita la página WDM
2. Sistema: Procesa la petición
3. Usuario: Hace click en el enlace “doctor”
4. Sistema: Procesa la petición y presenta la página
5. Usuario: Selecciona la opción “ver consultas no resueltas”
6. Sistema: Procesa la petición y presenta la página con el listado de consultas
7. Usuario: Selecciona una zona del cuerpo donde se ha realizado una consulta
8. Sistema: Procesa la petición y presenta el listado de consultas filtrado por esa zona
9. Usuario: Hace click en el icono de “ver detalle” de una consulta
10. Sistema: Procesa la petición y presenta la página de detalle de una consulta
11. Usuario: Selecciona la opción “Generar consulta relacionada”
12. Sistema: Procesa la petición y presenta el formulario para consulta relacionada
13. Usuario: Rellena los campos y envía los datos
14. Sistema: Valida los datos
15. Sistema: Procesa la petición y crea una consulta relacionada
16. Sistema: Redirige al listado de consultas

**Extensiones**

3.1 Si el usuario no ha hecho login o si lo ha hecho pero no es un usuario (es doctor o admin), se presenta la página de login.

14.1 Si los datos no son válidos, se vuelve al paso 6.

14.2 Muestra mensajes de error, en caso de que haya.

**Caso de uso 16.** Ver listado de usuarios

<b>Objetivo</b>	Ver el listado de usuarios de la plataforma
<b>Autor</b>	Doctor

**Precondiciones:** El doctor está logeado en el sistema

**Procedimiento**

1. Usuario: Visita la página WDM
2. Sistema: Procesa la petición
3. Usuario: Hace click en el enlace “doctor”
4. Sistema: Procesa la petición y presenta la página
5. Usuario: Selecciona la opción “ver usuarios”
6. Sistema: Procesa la petición y presenta la página
7. Usuario: Visualiza la lista de usuarios

**Extensiones**

3.1 Si el doctor no ha hecho login o si lo ha hecho pero no es un doctor (es usuario o admin), se presenta la página de login.

**Caso de uso 17.** Ver detalle de usuario

<b>Objetivo</b>	Ver en detalle la información de un usuario
-----------------	---

<b>Autor</b>	Doctor
<b>Precondiciones:</b> El doctor está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “doctor”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver usuarios”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de usuarios</li> <li>7. Usuario: Hace click en el icono de “ver detalle” de un usuario</li> <li>8. Sistema: Procesa la petición y presenta la página de detalle</li> <li>9. Usuario: Visualiza los detalles del usuario</li> </ol>	
<b>Extensiones</b> <p>3.1 Si el doctor no ha hecho login o si lo ha hecho pero no es un doctor (es usuario o admin), se presenta la página de login.</p>	

<b>Caso de uso 18.</b> Editar usuario	
<b>Objetivo</b>	Editar la información de un usuario
<b>Autor</b>	Doctor
<b>Precondiciones:</b> El doctor está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “doctor”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver usuarios”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de usuarios</li> <li>7. Usuario: Hace click en el icono de “editar” de un usuario</li> <li>8. Sistema: Procesa la petición y presenta la formulario de edición</li> <li>9. Usuario: Rellena los campos y envía los datos</li> <li>10. Sistema: Valida los datos</li> <li>11. Sistema: Procesa la información y actualiza los datos del usuario</li> <li>12. Sistema: Redirige a la página de detalle de usuario</li> </ol>	
<b>Extensiones</b> <p>3.1 Si el doctor no ha hecho login o si lo ha hecho pero no es un doctor (es usuario o admin), se presenta la página de login.</p> <p>10.1 Si los datos no son válidos, se vuelve al paso 6.</p> <p>10.2 Muestra mensajes de error, en caso de que haya.</p>	



<b>Caso de uso 19.</b> Crear usuario	
<b>Objetivo</b>	Crear un nuevo usuario
<b>Autor</b>	Doctor
<b>Precondiciones:</b> El doctor está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “doctor”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver usuarios”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de usuarios</li> <li>7. Usuario: Hace click en el icono de “Nuevo usuario”</li> <li>8. Sistema: Procesa la petición y presenta la formulario de creación de usuario</li> <li>9. Usuario: Rellena los campos y envía los datos</li> <li>10. Sistema: Valida los datos</li> <li>11. Sistema: Procesa la información y un usuario</li> <li>12. Sistema: Redirige al listado de usuarios</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el doctor no ha hecho login o si lo ha hecho pero no es un doctor (es usuario o admin), se presenta la página de login.</li> <li>10.1 Si los datos no son válidos, se vuelve al paso 6.</li> <li>10.2 Muestra mensajes de error, en caso de que haya.</li> </ol>	

<b>Caso de uso 20.</b> Borrar usuario	
<b>Objetivo</b>	Borrar un usuario
<b>Autor</b>	Doctor
<b>Precondiciones:</b> El doctor está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “doctor”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver usuarios”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de usuarios</li> <li>7. Usuario: Hace click en el icono de “borrar” de un usuario</li> <li>8. Sistema: Procesa la petición y borra al usuario</li> <li>9. Sistema: Actualiza la información del listado de usuarios</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el doctor no ha hecho login o si lo ha hecho pero no es un doctor (es usuario o admin), se presenta la página de login.</li> </ol>	

### 14.2.4 Administrador

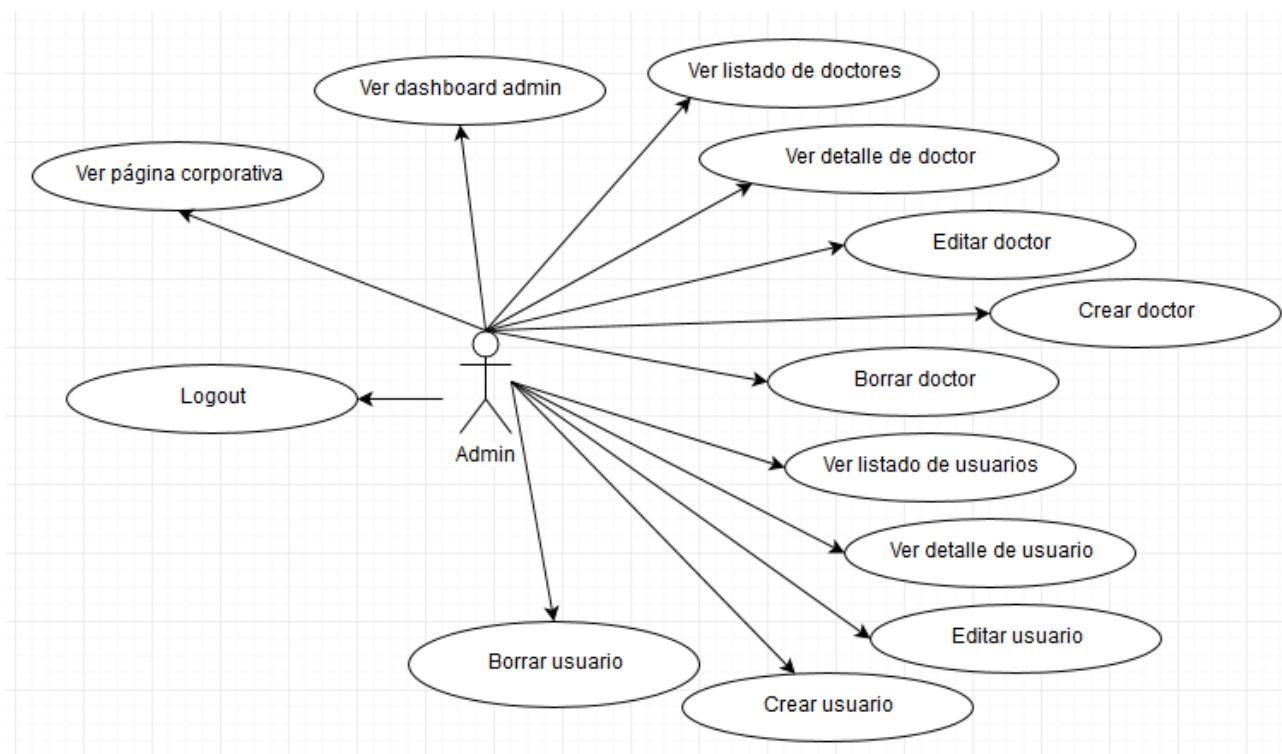


Ilustración 77: Casos de uso de administrador.

<b>Caso de uso 21. Logout</b>	
<b>Objetivo</b>	Un usuario autenticado se desconecta del sistema
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>11. Usuario: Visita la página WDM</li> <li>12. Sistema: Procesa la petición</li> <li>13. Usuario: Selecciona la opción “logout”</li> <li>14. Sistema: Procesa la petición y desconecta al usuario del sistema</li> <li>15. Sistema: Redirige a la página principal</li> </ol>	
<b>Extensiones</b>	

<b>Caso de uso 22. Ver página corporativa</b>	
<b>Objetivo</b>	Ver la información de la página corporativa
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	

<b>Procedimiento</b>	
1. Usuario:	Visita la página WDM
2. Sistema:	Procesa la petición
3. Usuario:	Navega por la página corporativa
<b>Extensiones</b>	

<b>Caso de uso 23.</b> Ver listado de usuarios	
<b>Objetivo</b>	Ver el listado de usuarios de la plataforma
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “Admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver usuarios”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de usuarios</li> <li>7. Usuario: Visualiza la lista de usuarios</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se presenta la página de login de admin.</li> </ol>	

<b>Caso de uso 24.</b> Ver detalle de usuario	
<b>Objetivo</b>	Ver en detalle la información de un usuario
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver usuarios”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de usuarios</li> <li>7. Usuario: Hace click en el icono de “ver detalle” de un usuario</li> <li>8. Sistema: Procesa la petición y presenta la página de detalle</li> <li>9. Usuario: Visualiza los detalles del usuario</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se presenta la página de login de admin.</li> </ol>	

<b>Caso de uso 25.</b> Editar usuario	
<b>Objetivo</b>	Editar la información de un usuario
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver usuarios”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de usuarios</li> <li>7. Usuario: Hace click en el icono de “editar” de un usuario</li> <li>8. Sistema: Procesa la petición y presenta la formulario de edición</li> <li>9. Usuario: Rellena los campos y envía los datos</li> <li>10. Sistema: Valida los datos</li> <li>11. Sistema: Procesa la información y actualiza los datos del usuario</li> <li>12. Sistema: Redirige a la página de detalle de usuario</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se presenta la página de login de admin.</li> <li>10.1 Si los datos no son válidos, se vuelve al paso 6.</li> <li>10.2 Muestra mensajes de error, en caso de que haya.</li> </ol>	

<b>Caso de uso 26.</b> Crear usuario	
<b>Objetivo</b>	Crear un nuevo usuario
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver usuarios”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de usuarios</li> <li>7. Usuario: Hace click en el icono de “Nuevo usuario”</li> <li>8. Sistema: Procesa la petición y presenta la formulario de creación de usuario</li> <li>9. Usuario: Rellena los campos y envía los datos</li> <li>10. Sistema: Valida los datos</li> <li>11. Sistema: Procesa la información y un usuario</li> <li>12. Sistema: Redirige al listado de usuarios</li> </ol>	
<b>Extensiones</b>	

3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se presenta la página de login de admin.

10.1 Si los datos no son válidos, se vuelve al paso 6.

10.2 Muestra mensajes de error, en caso de que haya.

<b>Caso de uso 27. Borrar usuario</b>	
<b>Objetivo</b>	Borrar un usuario
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver usuarios”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de usuarios</li> <li>7. Usuario: Hace click en el icono de “borrar” de un usuario</li> <li>8. Sistema: Procesa la petición y borra al usuario</li> <li>9. Sistema: Actualiza la información del listado de usuarios</li> </ol>	
<b>Extensiones</b> <p>3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se presenta la página de login de admin.</p>	

<b>Caso de uso 28. Ver listado de doctores</b>	
<b>Objetivo</b>	Ver el listado de doctores de la plataforma
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “Admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver doctores”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de doctores</li> <li>7. Usuario: Visualiza la lista de doctores</li> </ol>	
<b>Extensiones</b> <p>3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se presenta la página de login de admin.</p>	

<b>Caso de uso 29.</b> Ver detalle de doctor	
<b>Objetivo</b>	Ver en detalle la información de un doctor
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver doctores”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de doctores</li> <li>7. Usuario: Hace click en el icono de “ver detalle” de un doctor</li> <li>8. Sistema: Procesa la petición y presenta la página de detalle</li> <li>9. Usuario: Visualiza los detalles del doctor</li> </ol>	
<b>Extensiones</b> <p>3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se presenta la página de login de admin.</p>	

<b>Caso de uso 30.</b> Editar doctor	
<b>Objetivo</b>	Editar la información de un doctor
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver doctores”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de doctores</li> <li>7. Usuario: Hace click en el icono de “editar” de un doctor</li> <li>8. Sistema: Procesa la petición y presenta la formulario de edición</li> <li>9. Usuario: Rellena los campos y envía los datos</li> <li>10. Sistema: Valida los datos</li> <li>11. Sistema: Procesa la información y actualiza los datos del doctor</li> <li>12. Sistema: Redirige a la página de detalle de doctor</li> </ol>	
<b>Extensiones</b> <p>3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se presenta la página de login de admin.</p> <p>10.1 Si los datos no son válidos, se vuelve al paso 6.</p> <p>10.2 Muestra mensajes de error, en caso de que haya.</p>	

<b>Caso de uso 31.</b> Crear doctor	
<b>Objetivo</b>	Crear un nuevo doctor
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver doctores”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de doctores</li> <li>7. Usuario: Hace click en el icono de “Nuevo doctor”</li> <li>8. Sistema: Procesa la petición y presenta la formulario de creación de doctor</li> <li>9. Usuario: Rellena los campos y envía los datos</li> <li>10. Sistema: Valida los datos</li> <li>11. Sistema: Procesa la información y un doctor</li> <li>12. Sistema: Redirige al listado de doctor</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se presenta la página de login de admin.</li> <li>10.1 Si los datos no son válidos, se vuelve al paso 6.</li> <li>10.2 Muestra mensajes de error, en caso de que haya.</li> </ol>	

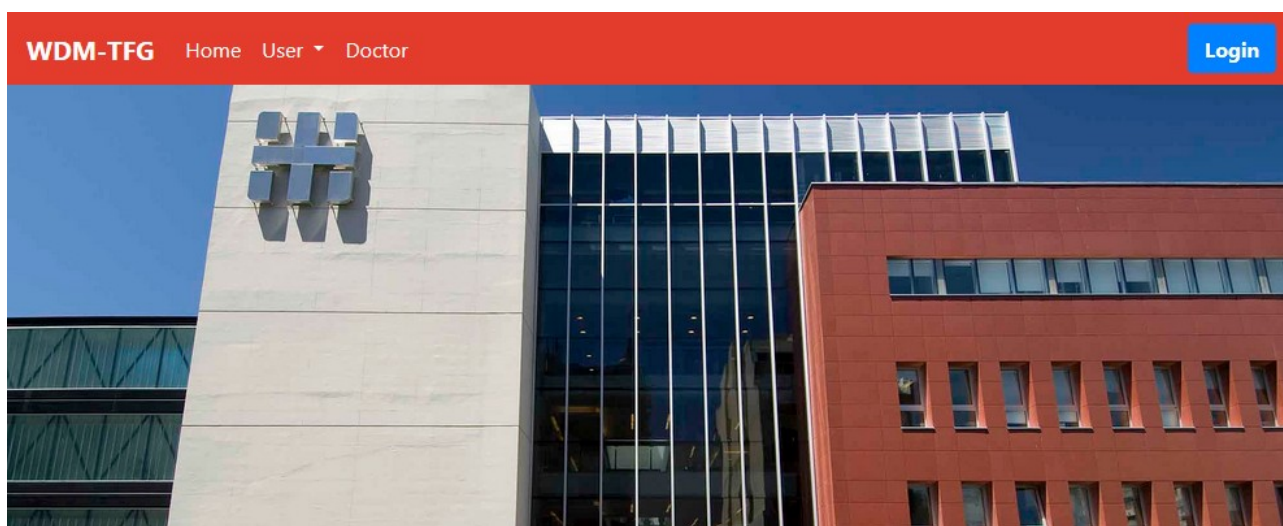
<b>Caso de uso 32.</b> Borrar doctor	
<b>Objetivo</b>	Borrar un doctor
<b>Autor</b>	Admin
<b>Precondiciones:</b> El admin está logeado en el sistema	
<b>Procedimiento</b> <ol style="list-style-type: none"> <li>1. Usuario: Visita la página WDM</li> <li>2. Sistema: Procesa la petición</li> <li>3. Usuario: Hace click en el enlace “admin”</li> <li>4. Sistema: Procesa la petición y presenta la página</li> <li>5. Usuario: Selecciona la opción “ver doctor”</li> <li>6. Sistema: Procesa la petición y presenta la página de listado de doctores</li> <li>7. Usuario: Hace click en el icono de “borrar” de un doctor</li> <li>8. Sistema: Procesa la petición y borra al doctor</li> <li>9. Sistema: Actualiza la información del listado de doctores</li> </ol>	
<b>Extensiones</b> <ol style="list-style-type: none"> <li>3.1 Si el admin no ha hecho login o si lo ha hecho pero no es un admin (es usuario o doctor), se</li> </ol>	

presenta la página de login de admin.

## 14.3 Manual de usuario

A continuación se presenta un manual de usuario que servirá para poder ver cómo utilizar las funcionalidades de la plataforma web. Este manual está dividido en cuatro partes, consistentes en la web corporativa (simple muestra de la misma), así como de las distintas áreas privadas (usuario, doctor y administrador).

### 14.3.1 Web corporativa

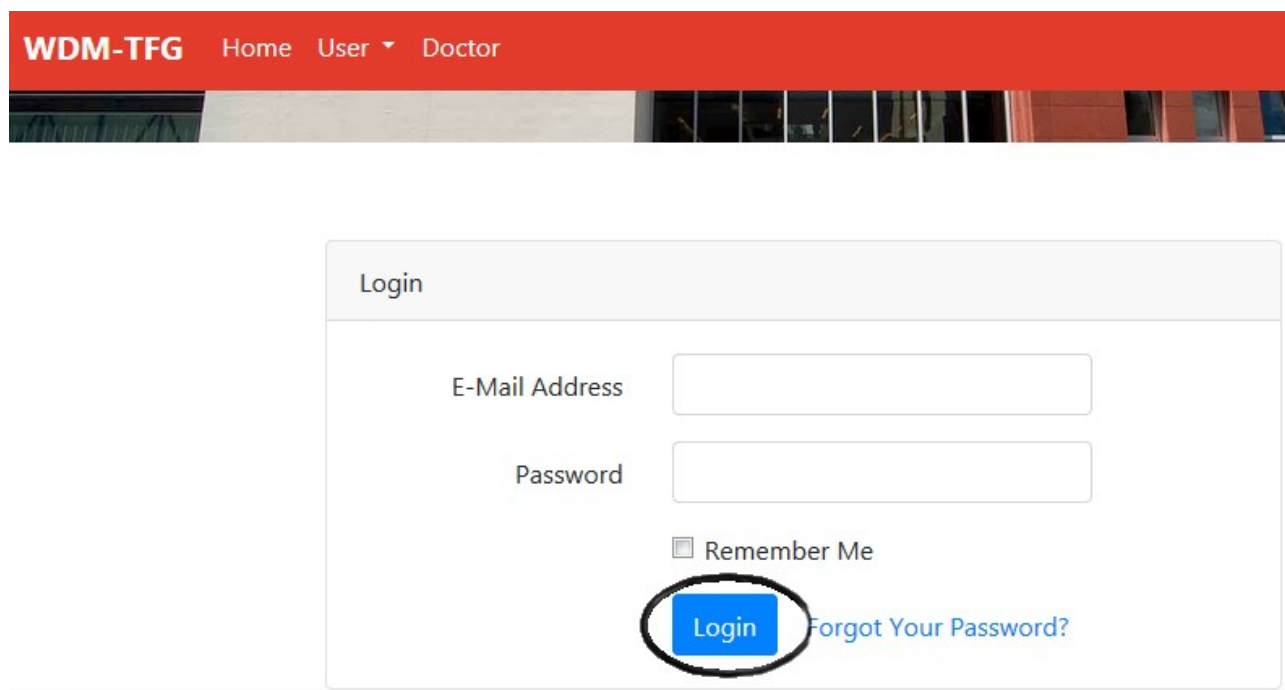


*Ilustración 78: Captura de la página principal corporativa*

Si el usuario accede a la URL de la plataforma web visible en la referencia [49], podrá ver la página principal. La página principal consiste en una página corporativa con información sobre el hospital.

Esta página sirve como punto de entrada a la aplicación, por lo que si el usuario desea acceder a las funcionalidades debe seleccionar la opción de menú que desee. En este caso, debe hacer login con credenciales de usuario o de doctor. En caso de estar logeado, podrá ver su nombre de usuario en el lugar donde está el botón de login.





The screenshot shows the top navigation bar of the WDM-TFG website with links for Home, User, and Doctor. Below the navigation bar is a login form titled "Login". The form contains two input fields: "E-Mail Address" and "Password". Below these fields is a checkbox labeled "Remember Me". At the bottom of the form is a blue "Login" button, which is circled in black, and a link labeled "Forgot Your Password?".

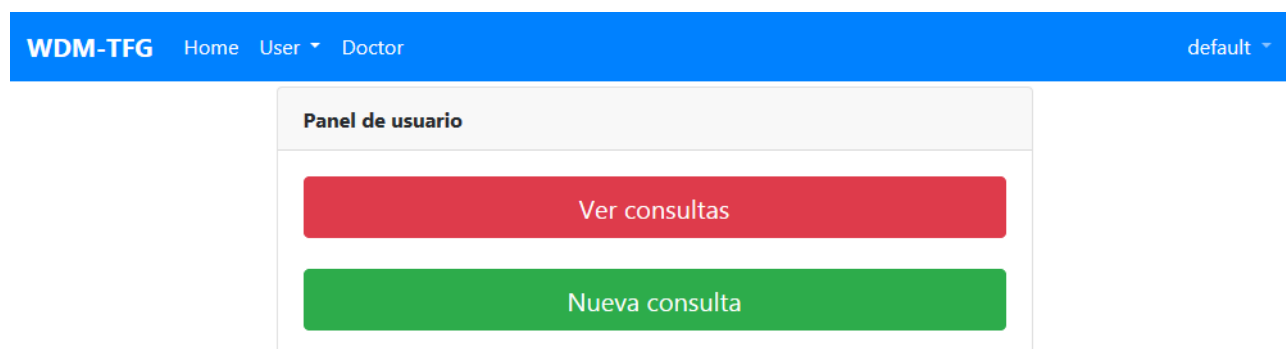
*Ilustración 79: Captura de la pantalla de login*

En caso de intentar acceder a algún área privada (usuario o doctor) sin haber hecho login o se pulsa el botón de “Login” de la página principal, aparecerá el formulario de login, tal y como se ve en la ilustración 79.

En este formulario el usuario deberá introducir un usuario y contraseña correctos y pulsar el botón “Login” indicado en la ilustración para acceder.

Será redirigido al panel de usuario o de doctor según las credenciales introducidas.

### 14.3.2 Área privada – usuario



The screenshot shows the user dashboard of the WDM-TFG website. The top navigation bar is blue and contains links for Home, User, and Doctor, along with a "default" dropdown menu. Below the navigation bar is a panel titled "Panel de usuario". Inside this panel are two buttons: a red button labeled "Ver consultas" and a green button labeled "Nueva consulta".

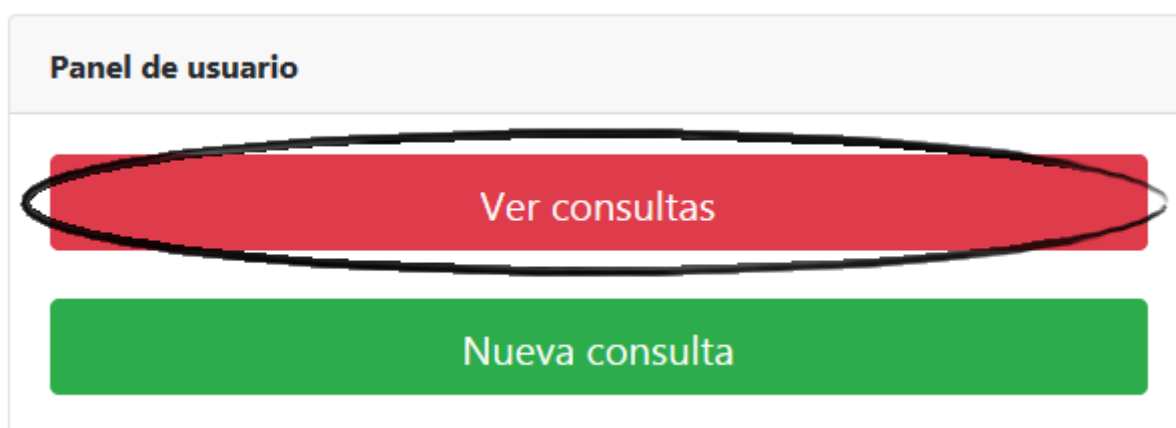
*Ilustración 80: Captura con el dashboard de usuario*



*Ilustración 81: Captura con la comprobación de que el usuario se ha logeado*

Una vez el usuario se ha logeado, si sus credenciales son de tipo usuario será redirigido al dashboard de usuario, representado en la ilustración 80. Este dashboard es la panel de usuario, donde podrá ver las opciones disponibles. Como se puede observar, la barra de navegación se ha cambiado a color azul, indicando que el usuario está en el área privada de tipo usuario (ilustración 81).

#### 14.3.2.1 Ver consultas



*Ilustración 82: Captura del botón de ver consultas*

El usuario puede ver las consultas que ha realizado si pulsa en el botón “Ver consultas” de su panel (ilustración 82).

**WDM-TFG** Home User ▾ Doctor

**Panel de usuario**

Listado de consultas

Nueva consulta

Filtrar Filtrar...

ID	Area	Created at	Acciones
1	1	2018-12-14 12:30:27	
2	5	2018-12-14 12:30:27	

Ilustración 83: Captura del listado de consultas de usuario

Al cambiar de página, el usuario podrá ver el listado de consultas que ha realizado. Este listado está agrupado por zonas del cuerpo. Esto quiere decir que de una determinada zona ha realizado varias consultas, tan solo aparecerá una de ellas (ilustración 83).

#### 14.3.2.2 Ver todas las consultas de una zona

Listado de consultas

Nueva consulta

Filtrar Filtrar...

ID	Area	Created at	Acciones
1	1	2018-12-14 12:30:27	

Ilustración 84: Captura del botón "ver más consultas" de una zona de consulta

Pulsando el botón indicado en la ilustración 84, es posible acceder al listado de consultas de la zona determinada (en el caso de la ilustración, la zona 1).

Panel de usuario

Consultas relacionadas

Nueva consulta

Filtrar

Filtrar...

ID	Related Query	Area	Created at	Result	Resolved	Acciones
1	1	3	2019-01-29 18:23:13	Sin riesgo	No	
3	1	3	2019-01-29 18:23:13	Sin riesgo	No	

Ilustración 85: Captura con el listado de consultas relacionadas por zona

Al acceder a esta página el usuario podrá ver todas las consultas realizadas sobre una zona (ilustración 85).

### 14.3.2.3 Ver detalle de una consultada

Consultas relacionadas

Nueva consulta

Filtrar

Filtrar...

ID	Related Query	Area	Created at	Result	Resolved	Acciones
1	1	3	2018-12-14 19:01:24	1	0	

Ilustración 86: Captura del botón "ver detalle" de una consulta


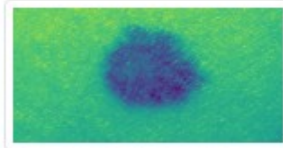
Una vez que el usuario se encuentra en el listado de “Consultas relacionadas”, donde puede ver todas las consultas que ha realizado de una zona, puede pulsar el botón “Ver detalle” de una consulta (ilustración 86).

Panel de usuario

Viendo detalle de consulta: 17

Image:

Image Mod:

User ID	Related Query	Zone	Created at	Result	Resolved
1	17	1	2019-01-30 15:00:40	Riesgo de melanoma elevado	No

Comment

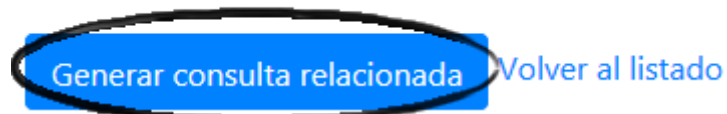
Ilustración 87: Captura de la página de detalle de consulta.

En la ilustración 87 se puede ver la pantalla de detalles de una consulta.

#### 14.3.2.4 Generar consulta relacionada

El término de consulta relacionada se refiere a una consulta que tiene relación con una consulta ya realizada. En la plataforma, quiere decir que será una consulta sobre una parte del cuerpo ya consultada.

Comentario:



*Ilustración 88: Captura del botón de "Generar consulta relacionada"*

Partiendo desde la pantalla de detalle de una consulta, visible en la ilustración 87, es necesario pulsar el botón “Generar consulta relacionada”, visible en la ilustración 88.

Una captura de pantalla de una aplicación web. En la parte superior hay una barra azul con el texto "WDM-TFG" y los enlaces "Home", "User" (con un menú desplegable) y "Doctor". Debajo, un panel gris claro con el título "Panel de usuario". El contenido principal es un formulario con el título "Creando consulta relacionada.". El formulario contiene: un campo "Subir imagen:" con un botón "Seleccionar archivo" y el texto "Ningún archivo seleccionado"; un campo "ID de la consulta relacionada:" con el valor "1"; un campo "Área de la imagen:" con el valor "3"; y dos botones al final: "Crear consulta" (azul) y "Volver atrás" (azul claro).

*Ilustración 89: Captura de la pantalla de generación de consulta relacionada*

Automáticamente se redirigirá la página hacia el formulario para crear una consulta relacionada (ilustración 89). En este caso lo único que se debe hacer es subir una imagen, pues el ID de la consulta relacionada y el área de la imagen (zona del cuerpo) están marcadas por defecto al ser una consulta relacionada.

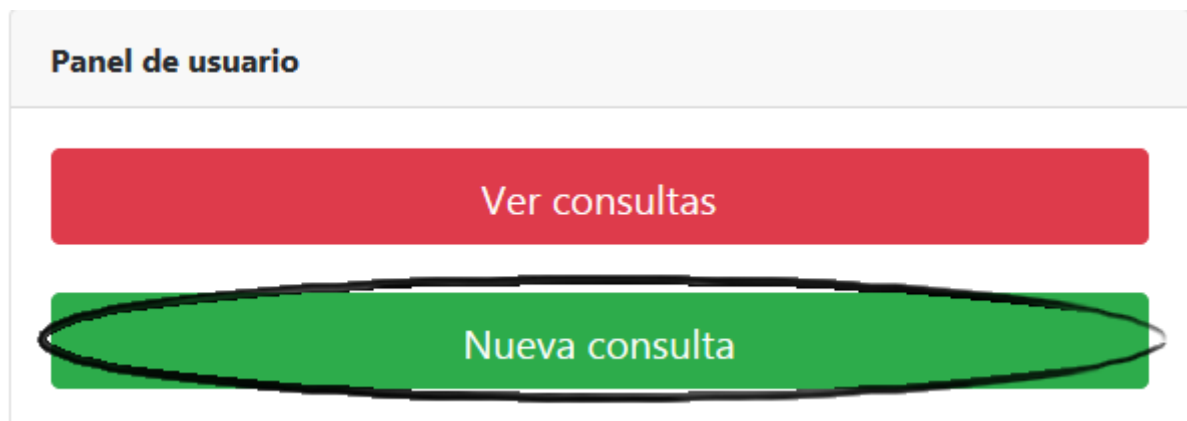


*Ilustración 90: Captura del botón para crear la consulta relacionada*

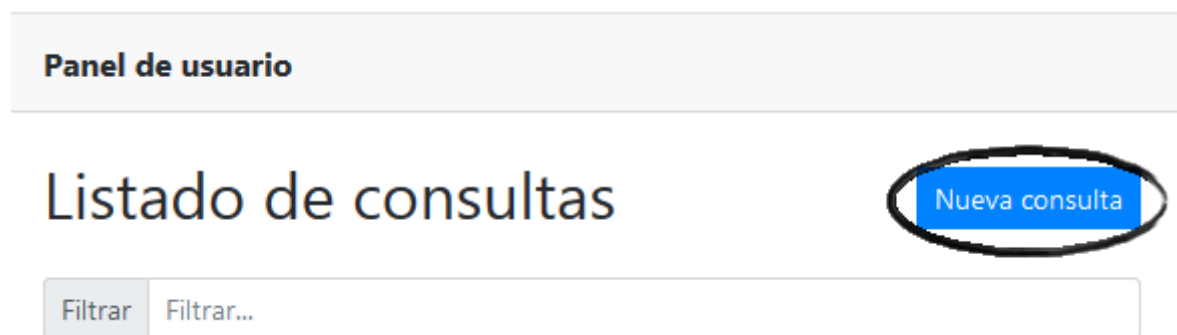
Para crear la consulta relacionada, una vez se ha seleccionado la imagen en el formulario de la ilustración 89, se debe pulsar el botón “Crear consulta”, tal y como se indica en la ilustración 90.

#### 14.3.2.5 Crear una nueva consulta

Cuando se crea una “nueva consulta”, se crea una consulta de una zona del cuerpo de la cual no se ha realizado ninguna consulta previamente.

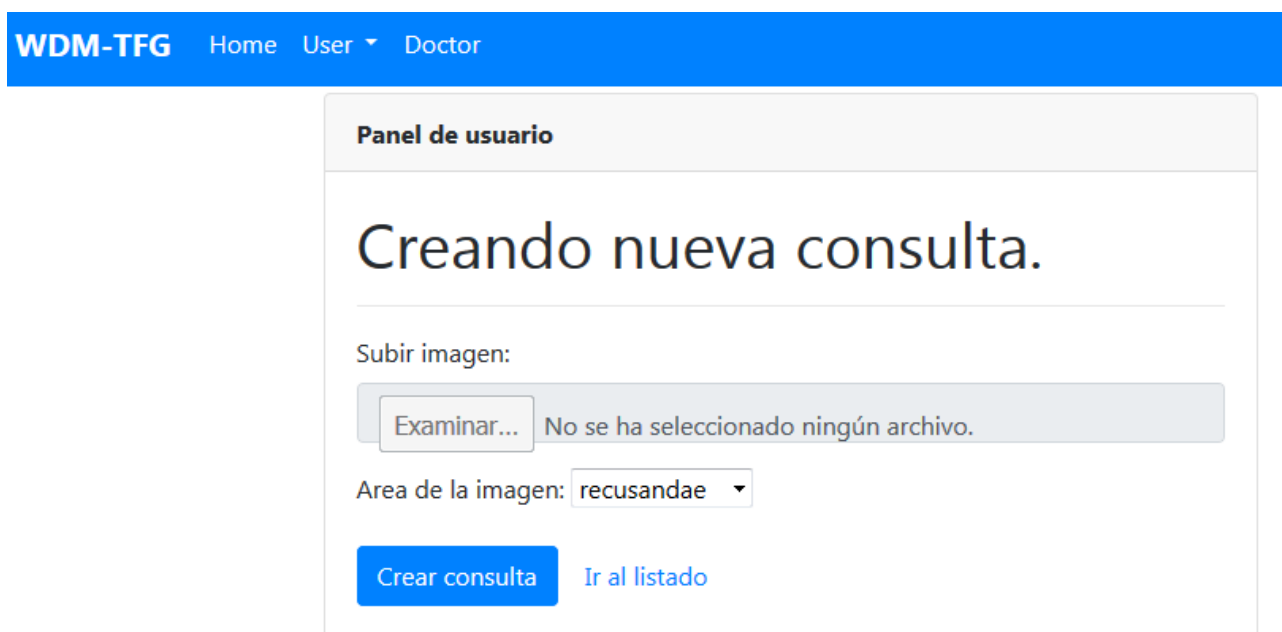


*Ilustración 91: Captura del botón "nueva consulta" del panel de usuario*



*Ilustración 92: Captura del botón "nueva consulta" del listado de consultas*

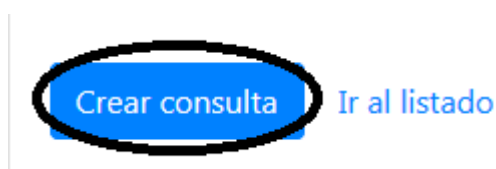
Pulsando en los botones de “nueva consulta” que se pueden ver en la ilustración 91 (panel de usuario) o en la ilustración 92 (listado de consultas), el usuario se dirigirá al formulario de creación de una nueva consulta.



The screenshot shows a web interface with a blue header bar containing the text 'WDM-TFG' and navigation links 'Home', 'User', and 'Doctor'. Below the header is a light gray box titled 'Panel de usuario'. Inside this box, the heading 'Creando nueva consulta.' is displayed. Underneath, there is a section labeled 'Subir imagen:' which includes a file upload button labeled 'Examinar...' and a message 'No se ha seleccionado ningún archivo.'. Below this is a dropdown menu labeled 'Area de la imagen:' with the value 'recusandae' selected. At the bottom of the form are two buttons: a blue button labeled 'Crear consulta' and a text link labeled 'Ir al listado'.

*Ilustración 93: Captura del formulario de nueva consulta*

En la ilustración 93 se puede ver el formulario de nueva consulta. En este caso, al tratarse de una consulta completamente nueva, se puede seleccionar una imagen para subir así como el área de la imagen (zona del cuerpo).



*Ilustración 94: Captura del botón de crear consulta*

Pulsando el botón de “Crear consulta” indicado en la ilustración 94, en el formulario de crear una nueva consulta, el sistema guardará la consulta.



### 14.3.3 Área privada – doctor

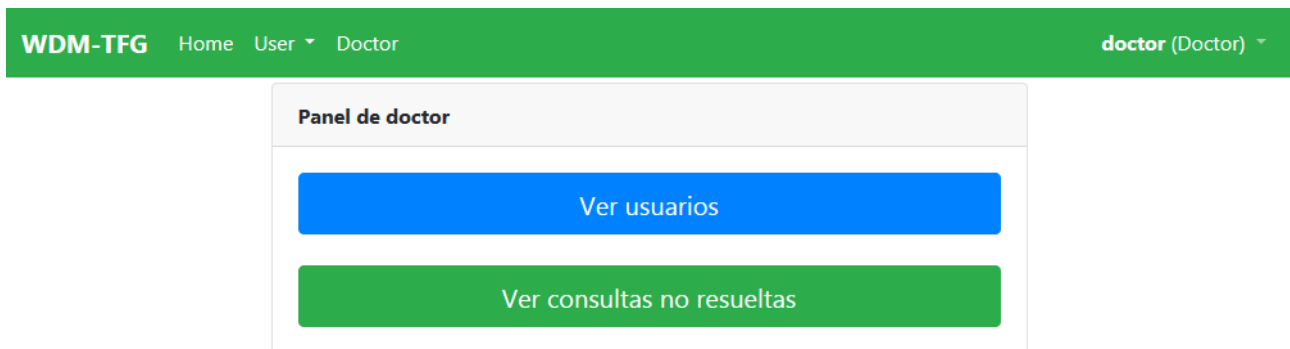


Ilustración 95: Captura del panel de doctor



Ilustración 96: Captura del detalle de doctor logeado

Una vez el usuario ha hecho login con unas credenciales de tipo doctor, el sistema le redirigirá al panel de doctor, que se puede ver en la ilustración 95. Como se puede observar, la barra de navegación pasa a ser de color verde, indicando que el usuario está en el área privada de doctor. Es posible también verificar este hecho ya que en el nombre de usuario aparece que es doctor entre paréntesis (ilustración 96).

#### 14.3.3.1 Ver listado de usuarios

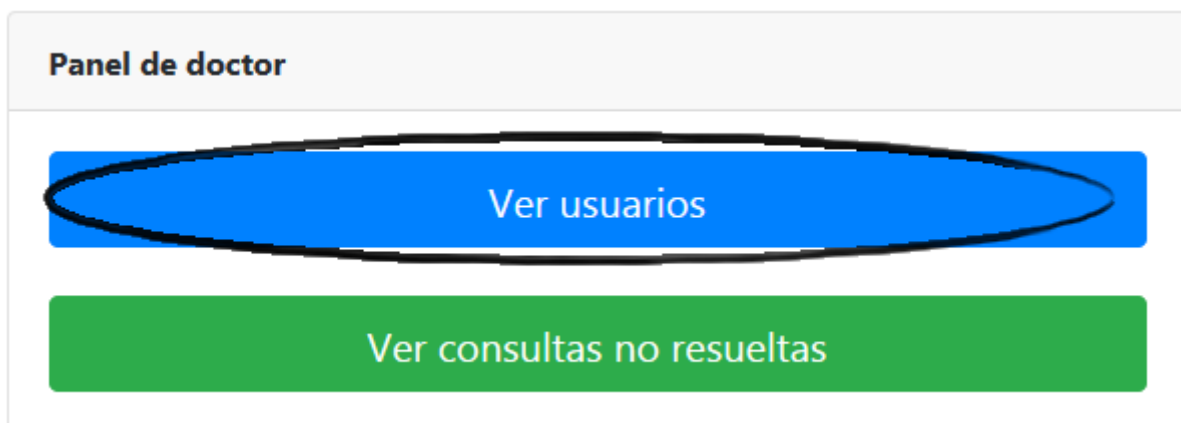


Ilustración 97: Captura del botón de "ver usuarios" del panel de doctor

En el panel de doctor, si el usuario pulsa al botón “Ver usuarios” indicado en la ilustración 97, se desplazará al listado de usuarios.

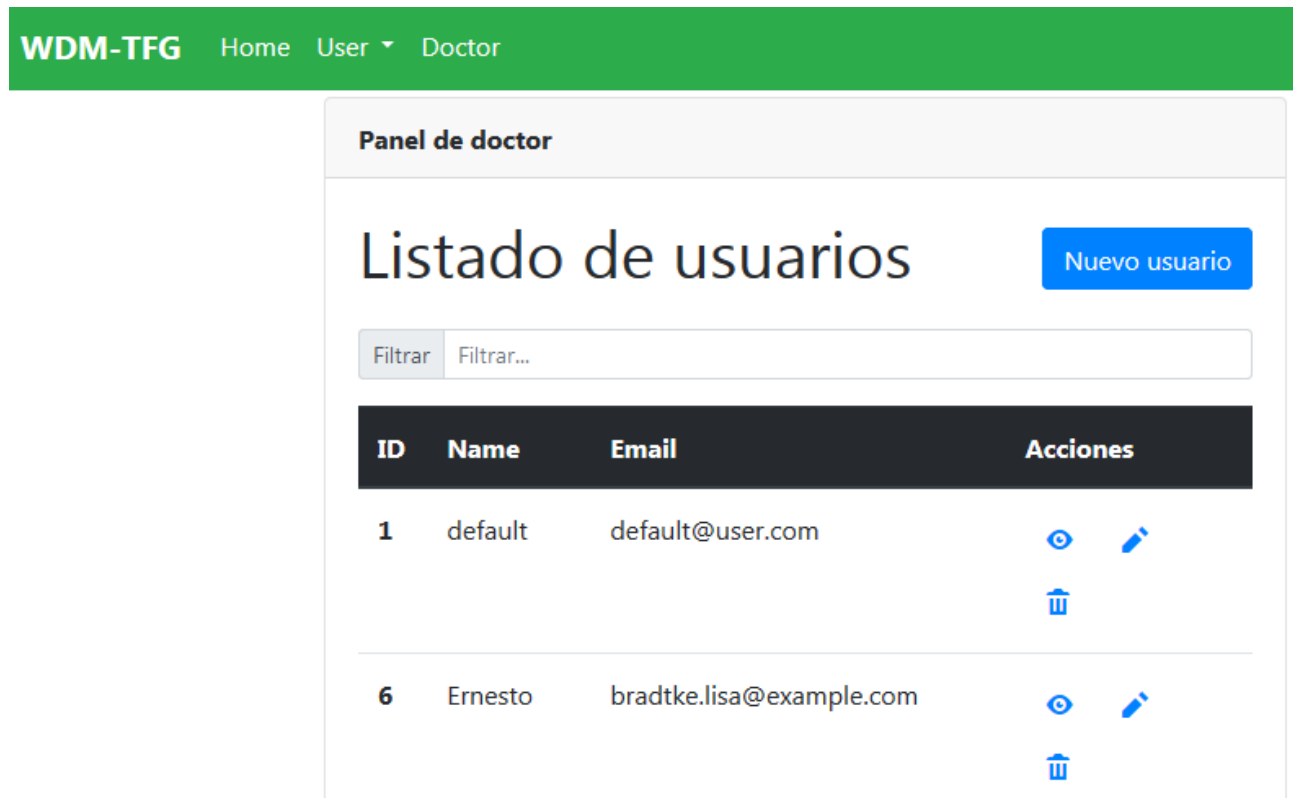


Ilustración 98: Captura de la página de listado de usuarios

El usuario podrá ver una página como la mostrada en la ilustración 98 con el listado de usuarios. En esta pantalla aparecen todos los usuarios que hay en el sistema y ofrece al doctor una serie de acciones a realizar sobre ellos.

A continuación se describen dichas acciones.

### 14.3.3.2 Ver detalle de usuario

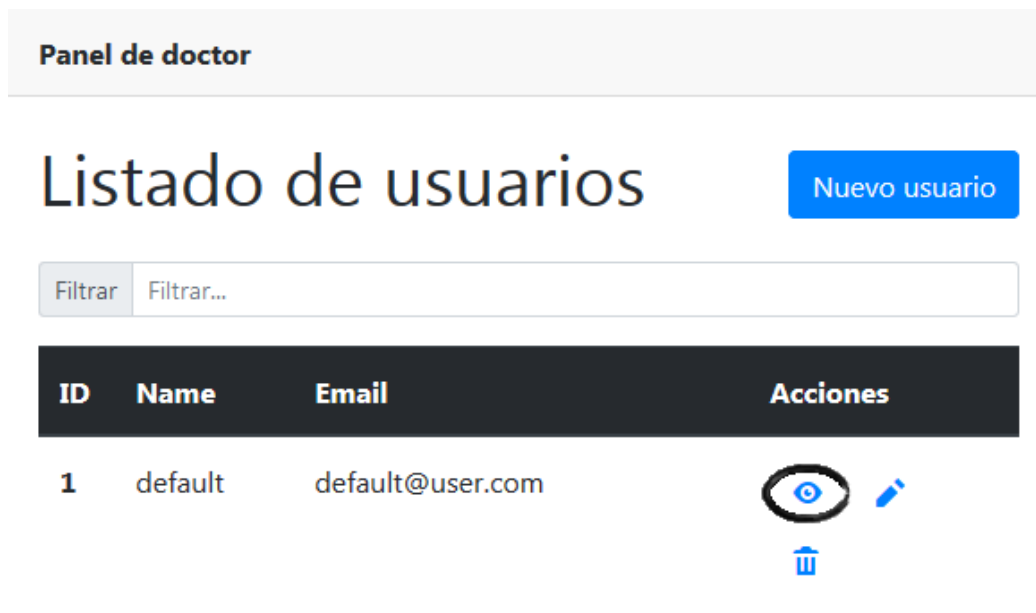


Ilustración 99: Captura del botón "Ver detalle" de un usuario

Es posible realizar tres acciones sobre un usuario creado. En este caso, es posible ver el icono "Ver detalles" de un usuario en la ilustración 99.

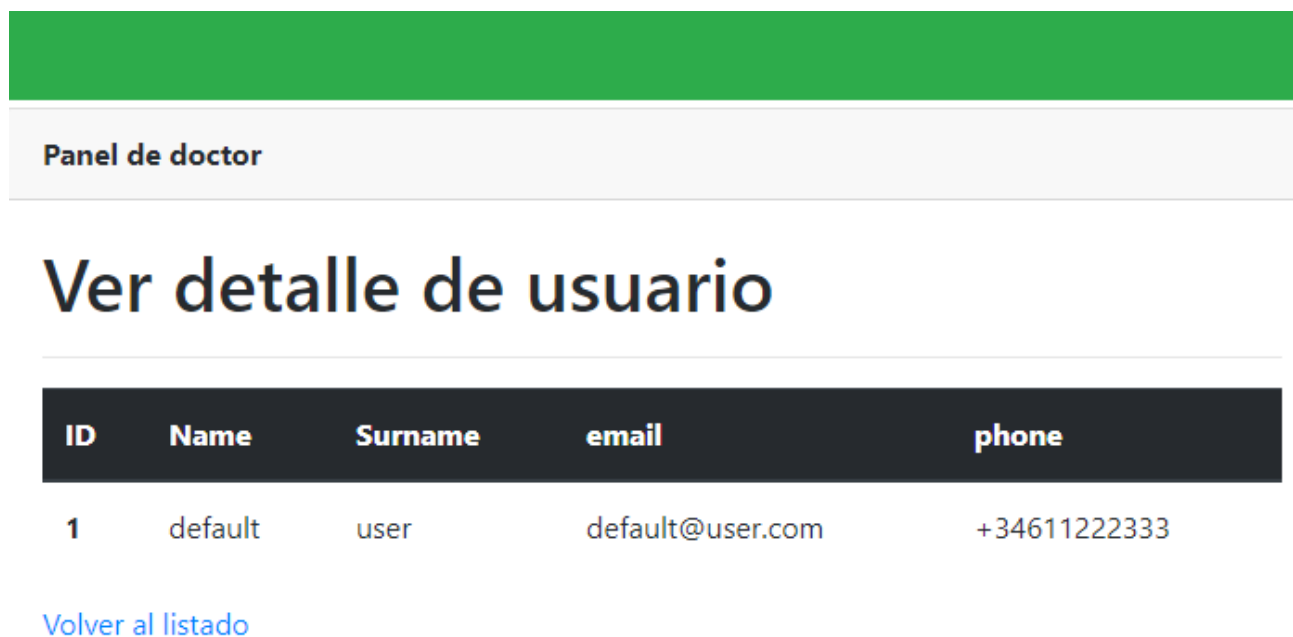
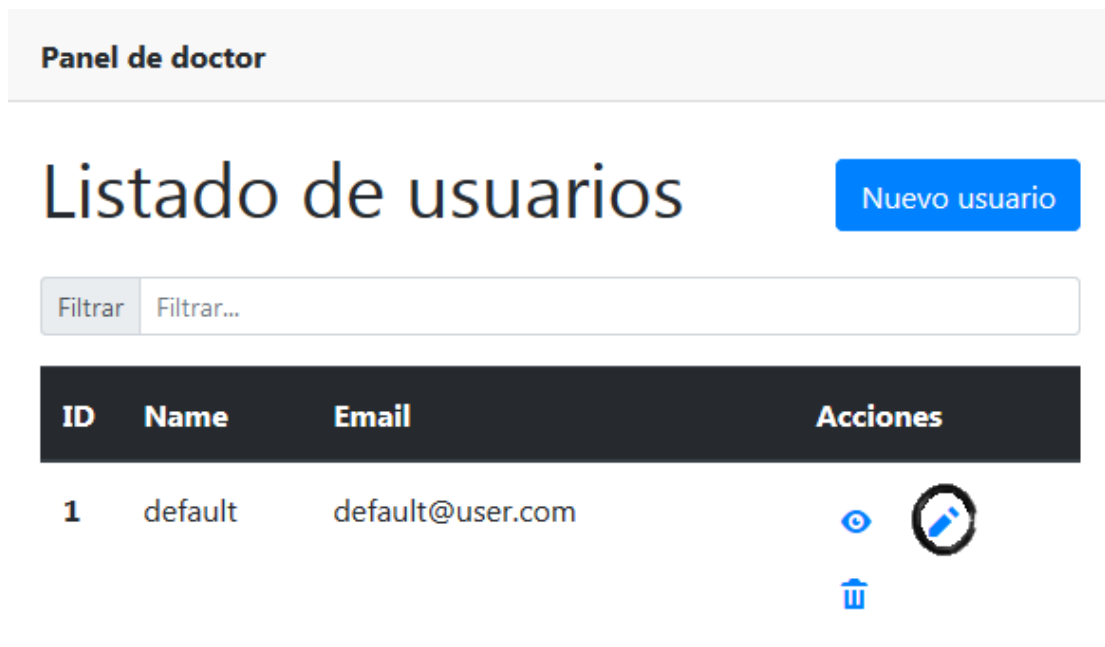


Ilustración 100: Captura de la página de detalle de usuario

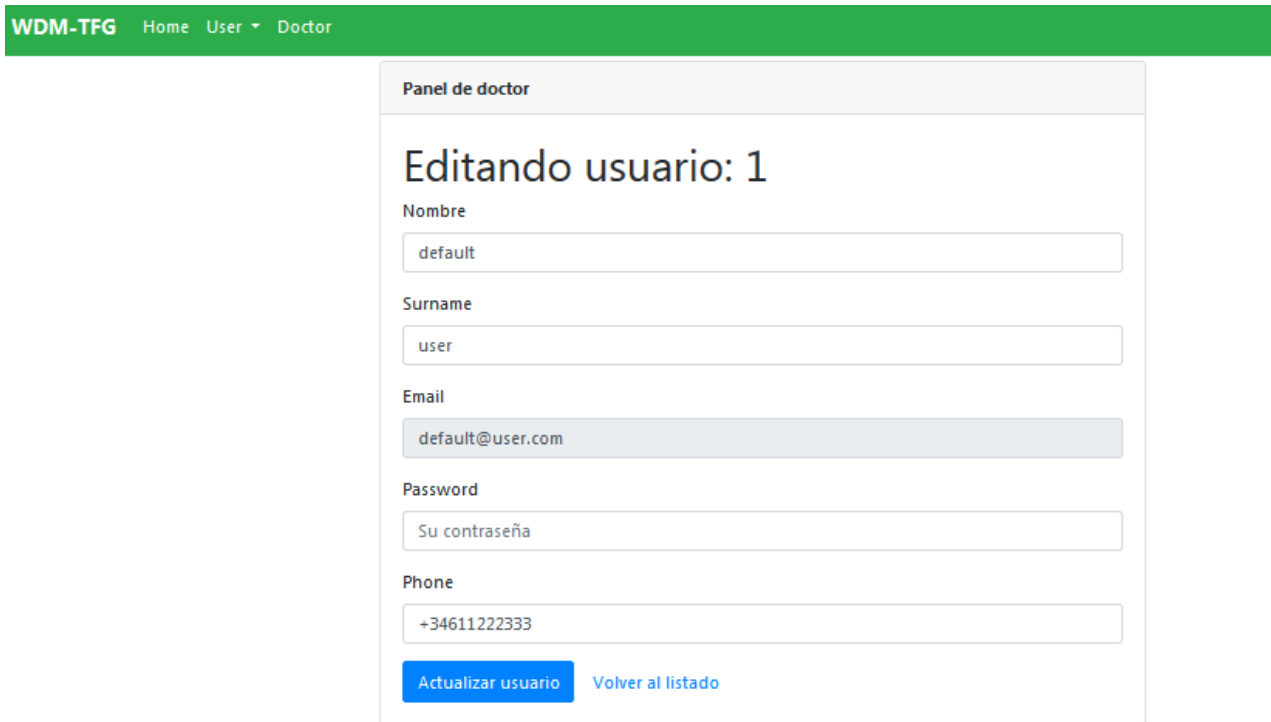
La página de detalle muestra la información completa de un usuario (ilustración 100).

### 14.3.3.3 Editar usuario



*Ilustración 101: Captura del botón de "editar" un usuario*

En la ilustración 101 es posible observar el botón de editar usuario.



WDM-TFG Home User ▾ Doctor

Panel de doctor

### Editando usuario: 1

Nombre

Surname

Email

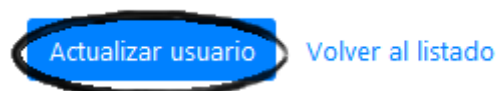
Password

Phone

[Actualizar usuario](#) [Volver al listado](#)

*Ilustración 102: Captura del formulario de edición de usuario*

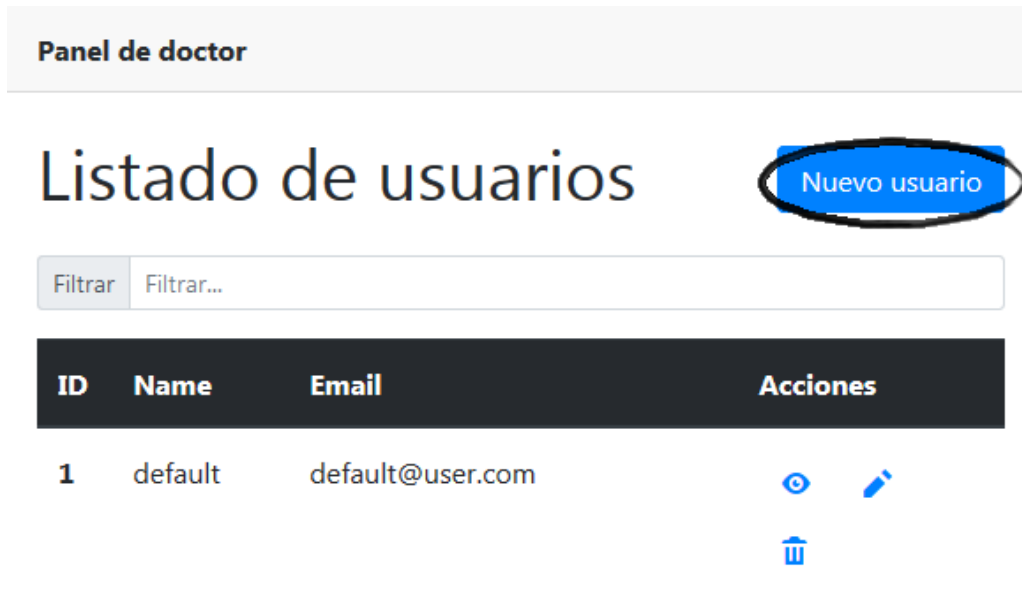
El usuario doctor puede editar todos los campos de un usuario a excepción del email (ilustración 102).



*Ilustración 103: Captura del botón de actualizar usuario*

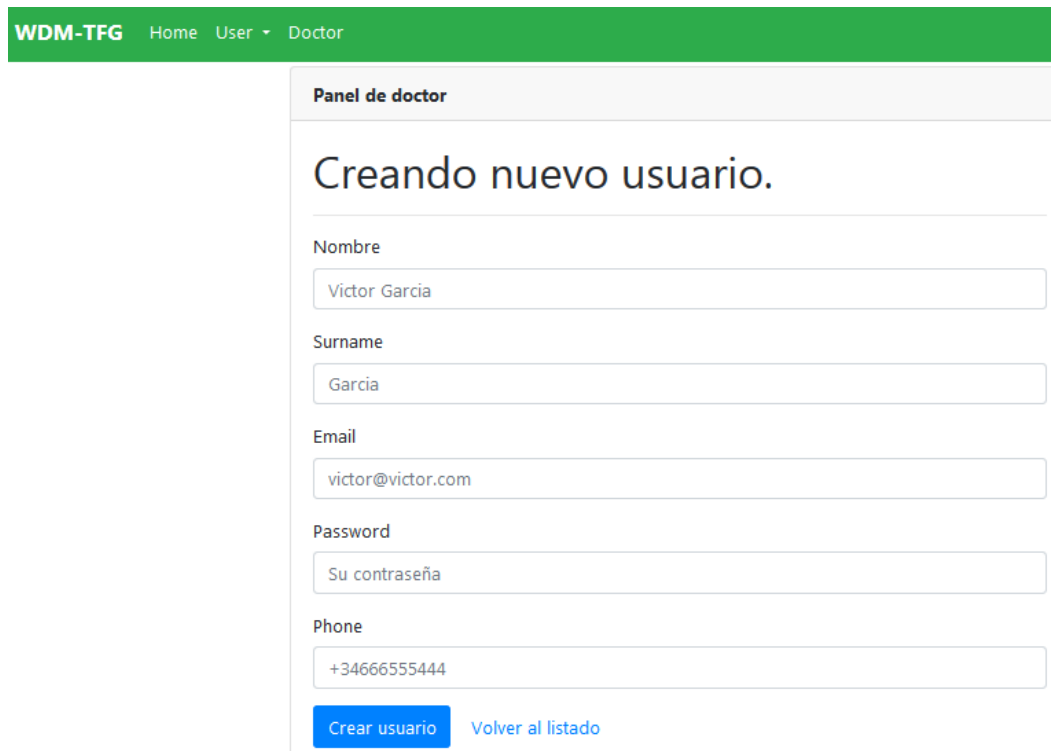
Pulsando el botón “Actualizar usuario” mostrado en la ilustración 103, en el formulario de edición de usuario, se guardarán los cambios realizados.

#### 14.3.3.4 Crear un usuario



*Ilustración 104: Captura del botón "nuevo usuario" del listado de usuarios*

Cuando el doctor se encuentra en el listado de usuarios, puede pulsar el botón “Nuevo usuario” para ser dirigido al formulario de creación de usuarios (ilustración 104). En dicho formulario podrá rellenar todos los campos correspondientes a la información de usuario y guardarlo en el sistema.



WDM-TFG Home User ▾ Doctor

Panel de doctor

## Creando nuevo usuario.

Nombre

Surname

Email

Password

Phone

[Crear usuario](#) [Volver al listado](#)

*Ilustración 105: Captura del formulario de creación de usuarios*

La ilustración 105 muestra el formulario de creación de usuarios.



*Ilustración 106: Captura del botón de crear usuario*

Al pulsar el botón “Crear usuario”, de la ilustración 106, el sistema guardará al nuevo usuario.

### 14.3.3.5 Borrar usuario

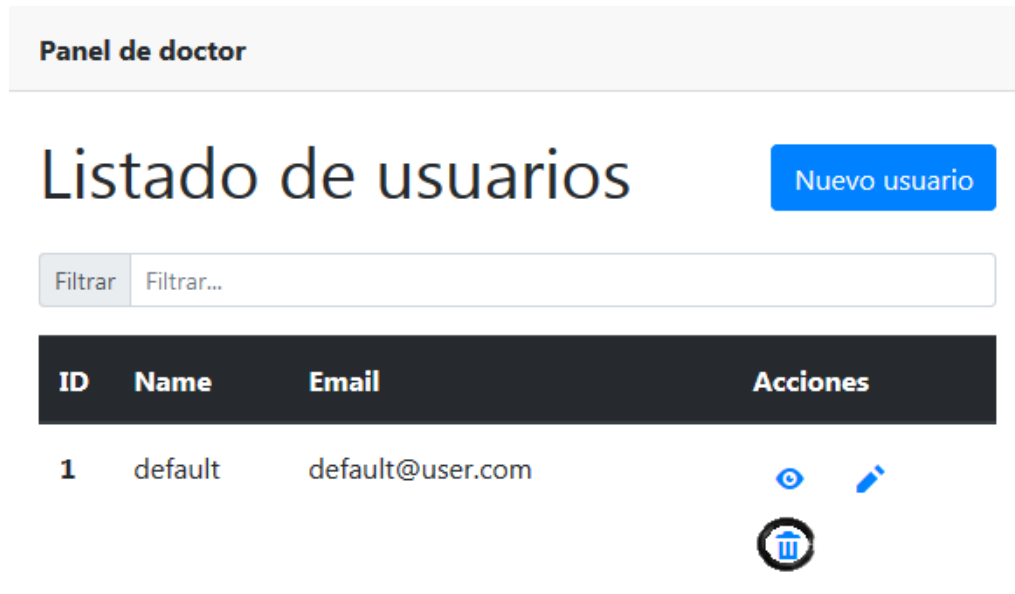


Ilustración 107: Captura del icono de borrar usuario

Pulsando el icono de “borrar usuario”, señalado en la ilustración 107 en el listado de usuarios, el sistema borrará el usuario indicado.

### 14.3.3.6 Ver consultas sin resolver

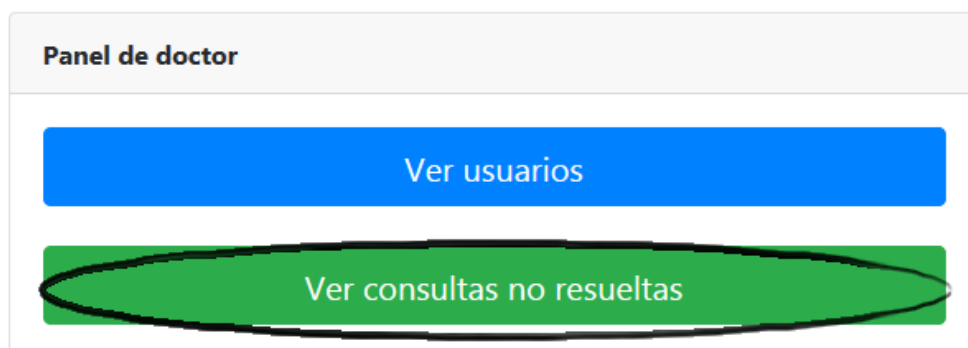


Ilustración 108: Captura del botón "Ver consultas no resueltas"

En la primera página del panel de doctor, es posible ver el botón “Ver consultas no resueltas” (ilustración 108).



Panel de doctor				
Listado de consultas				
<div>Filtrar <input type="text" value="Filtrar..."/></div>				
ID	Related Query	Area	Created at	Acciones
1	1	1	2018-12-14 12:30:27	
2	2	5	2018-12-14 12:30:27	
3	3	11	2018-12-14 12:30:27	

Ilustración 109: Captura del listado de consultas no resueltas

En este listado es posible ver las consultas que los usuarios han realizado en el sistema y no han sido resueltas (ilustración 109). Una consulta ha sido resuelta cuando un doctor ha podido revisarla y dejar un comentario.

#### 14.3.3.6 Ver detalle de consulta no resuelta

Panel de doctor				
Listado de consultas				
<div>Filtrar <input type="text" value="Filtrar..."/></div>				
ID	Related Query	Area	Created at	Acciones
1	1	1	2018-12-14 12:30:27	

Ilustración 110: Captura del icono "ver detalle" de consulta no resuelta

En el listado de consultas no resueltas, es posible ver el icono de “ver detalle” de una de las consultas, tal y como se ve en la ilustración 110.

**Panel de doctor**

## Ver detalle de consulta: 17

Image:



Image Mod:



User ID	Related Query	Zone	Created at	Result	Resolved
1	17	1	2019-01-30 15:00:40	Se han detectado problemas	No

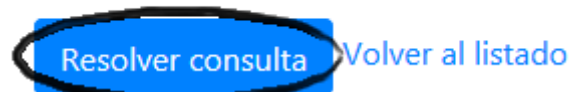
### Comment

Ilustración 111: Captura de los detalles de una consulta no resuelta

En la ilustración 111 se muestran los detalles de una consulta no resuelta. Cabe destacar como la consulta no resuelta no tiene ningún comentario.

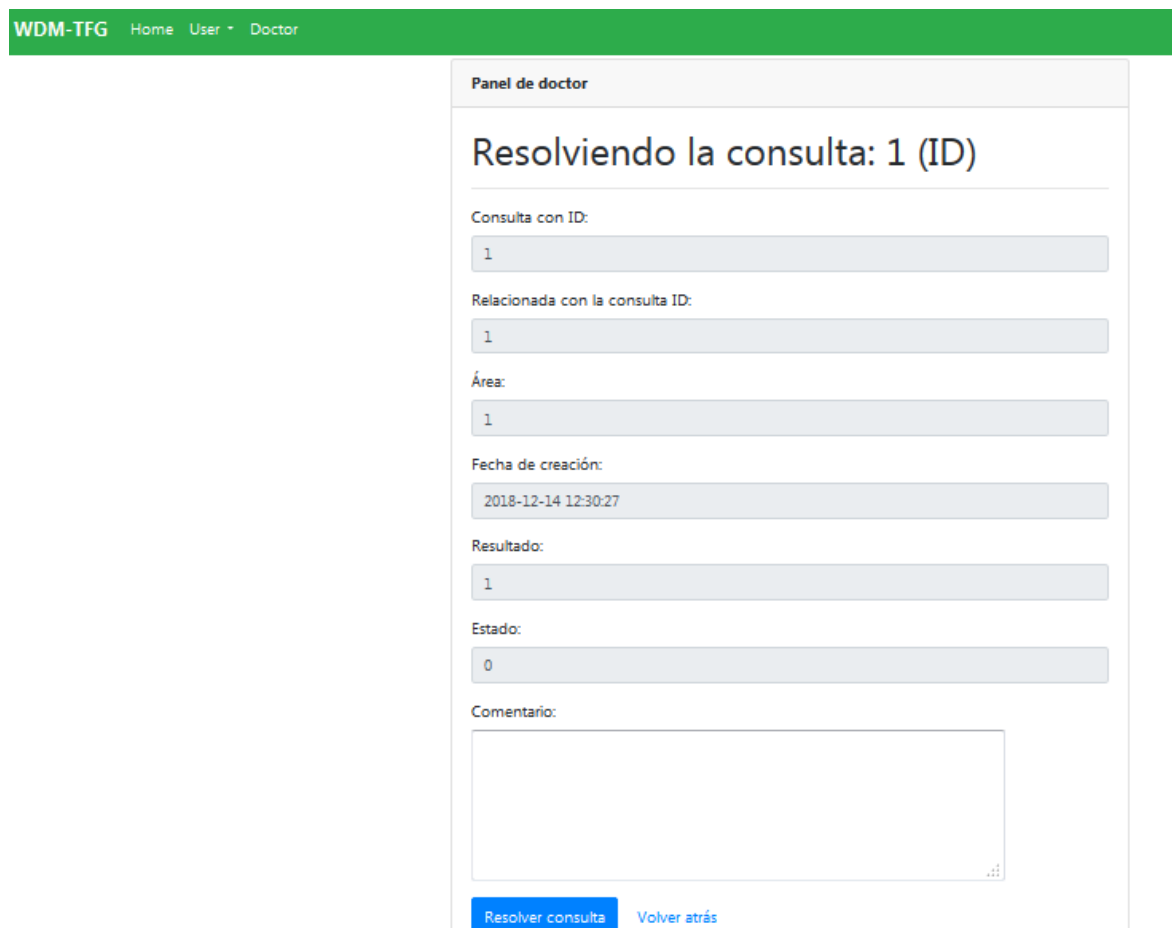
### 14.3.3.7 Resolver una consulta

Comentario:



*Ilustración 112: Captura del botón "Resolver consulta"*

En la misma página de los detalles de una consulta, aparece el botón “Resolver consulta”, como se ve en la ilustración 112.

A screenshot of a web application interface for a doctor's panel. The header is green with the text 'WDM-TFG Home User Doctor'. The main content area is titled 'Panel de doctor' and 'Resolviendo la consulta: 1 (ID)'. It contains several input fields: 'Consulta con ID:' (value: 1), 'Relacionada con la consulta ID:' (value: 1), 'Área:' (value: 1), 'Fecha de creación:' (value: 2018-12-14 12:30:27), 'Resultado:' (value: 1), and 'Estado:' (value: 0). There is a large text area for 'Comentario:'. At the bottom, there are two buttons: 'Resolver consulta' (blue) and 'Volver atrás' (blue).

*Ilustración 113: Captura del formulario de resolución de consulta*

En este caso, el único campo posible para editar es el comentario que el doctor pueda realizar al respecto de la consulta (ilustración 113).

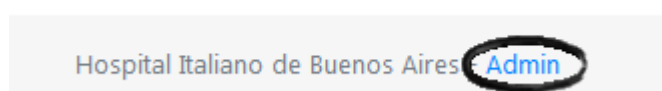


*Ilustración 114: Captura del botón de "Resolver consulta" del formulario de resolución de consulta*

Al escribir un comentario, el doctor puede pulsar el botón “Resolver consulta” para marcarla como resuelta (ilustración 114).

#### **14.3.4 Área privada – admin**

La plataforma web tiene un panel de administración. Para poder acceder a este panel de administración, además de disponer de unas credenciales de administrador, se debe realizar desde un apartado distinto al de un usuario y un doctor.



*Ilustración 115: Captura del link hacia el administrador en el footer*

En el pie de página (footer), es posible encontrar un link hacia el panel de administración (ilustración 115)

The screenshot shows the 'Panel de administrador' section of the WDM-TFG web application. It features an 'Admin Login' form with the following elements:

- E-Mail Address:** A text input field containing 'admin@admin.com'.
- Password:** A text input field.
- Remember Me:** A checkbox.
- Login as admin:** A blue button with the text 'Login as admin', which is circled in red.

*Ilustración 116: Captura del formulario de login de administrador*

Como se puede observar en la ilustración 116, el formulario de login de administrador es distinto que el de un usuario o doctor, ya que el proceso de login por parte de administradores está realizado en una sección independiente.

The screenshot shows the 'Panel de administrador' section of the WDM-TFG web application. It features two large buttons:

- Ver usuarios:** A blue button.
- Ver doctores:** A green button.

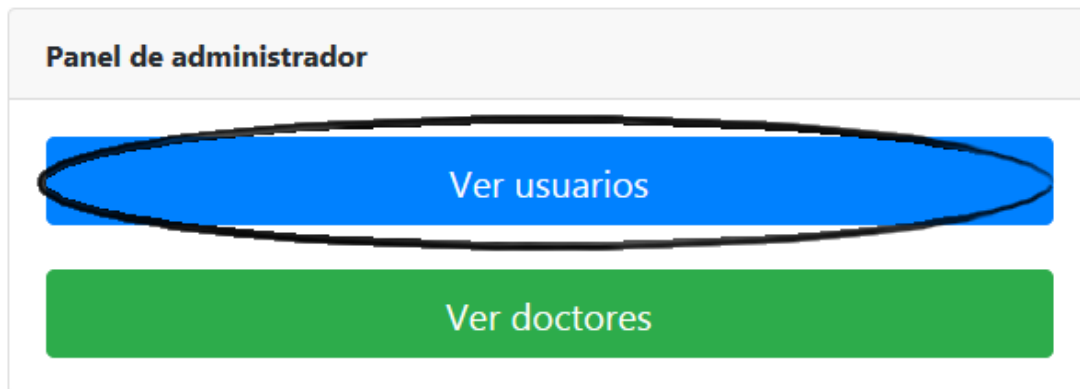
*Ilustración 117: Captura del panel de administrador*

The screenshot shows the header of the WDM-TFG web application. The navigation bar is dark grey. On the right side, the text 'admin (Admin)' is displayed, circled in red.

*Ilustración 118: Captura del detalle del nombre en el administrador*

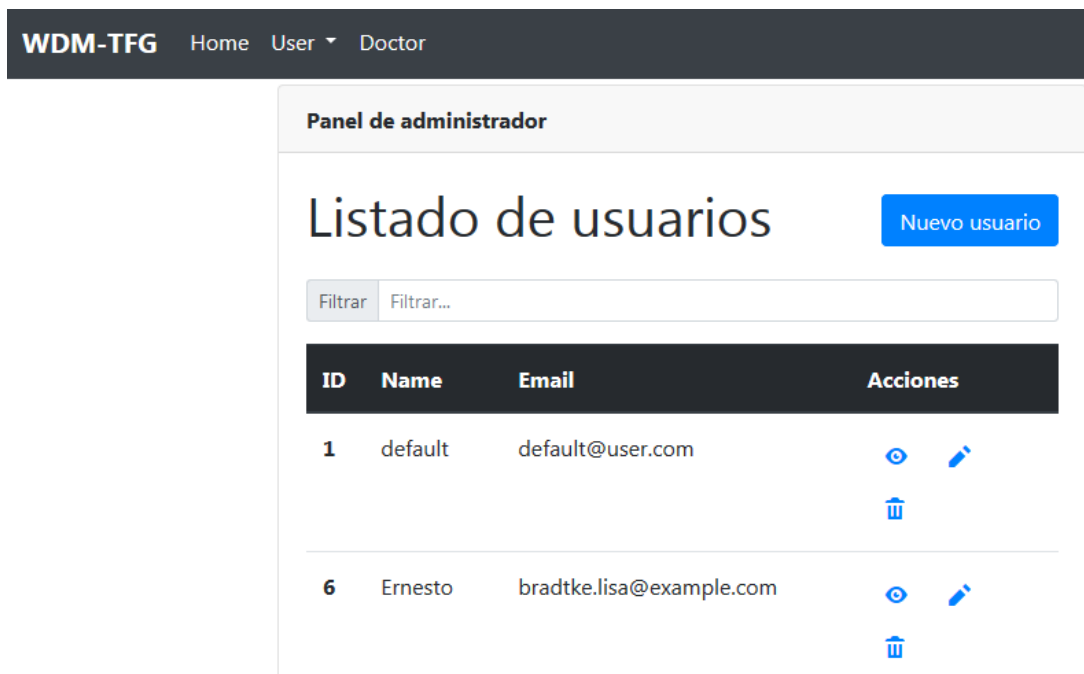
Al acceder al panel de administrador (ilustración 117), es posible observar como la barra de navegación pasa a ser de color negro (admin), así como se puede ver junto al nombre de admin que es de tipo “admin” (ilustración 118).

#### 14.3.4.1 Ver listado de usuarios



*Ilustración 119: Captura del botón "Ver usuarios" del panel de administrador*

En el panel de administrador, si el usuario pulsa al botón “Ver usuarios” indicado en la ilustración 119, se desplazará al listado de usuarios.

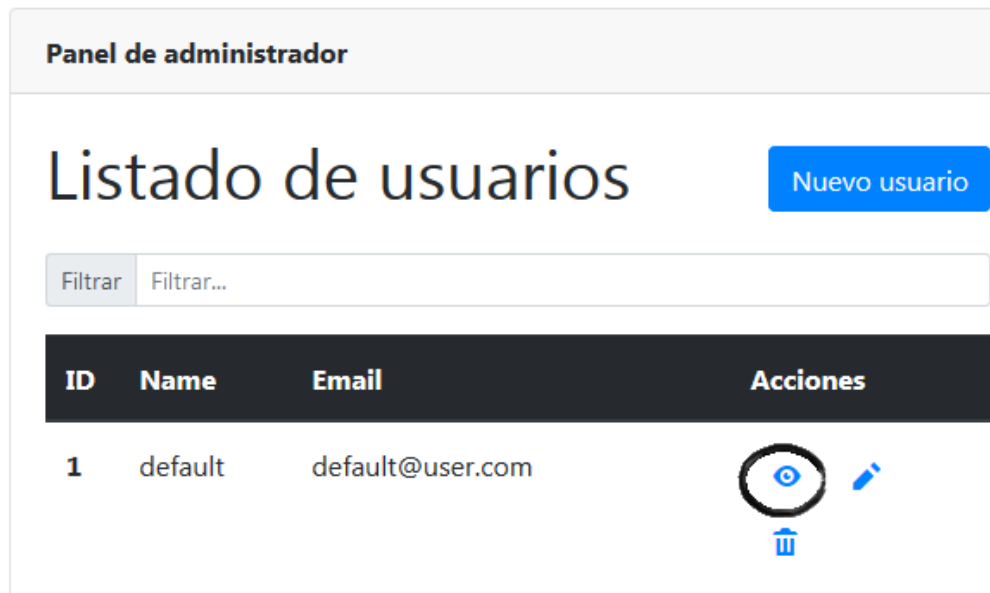


*Ilustración 120: Captura del listado de usuarios de administrador*

El administrador podrá ver una página como la mostrada en la ilustración 120 con el listado de usuarios. En esta pantalla aparecen todos los usuarios que hay en el sistema y ofrece al admin una serie de acciones a realizar sobre ellos.

A continuación se describen dichas acciones.

#### 14.3.4.2 Ver detalle de usuario



*Ilustración 121: Captura del icono de "ver detalle" de usuario*

Es posible realizar tres acciones sobre un usuario creado. En este caso, es posible ver el icono “Ver detalles” de un usuario en la ilustración 121.



Ilustración 122: Captura de los detalles de un usuario

La página de detalle muestra la información completa de un usuario (ilustración 122).

#### 14.3.4.3 Editar un usuario

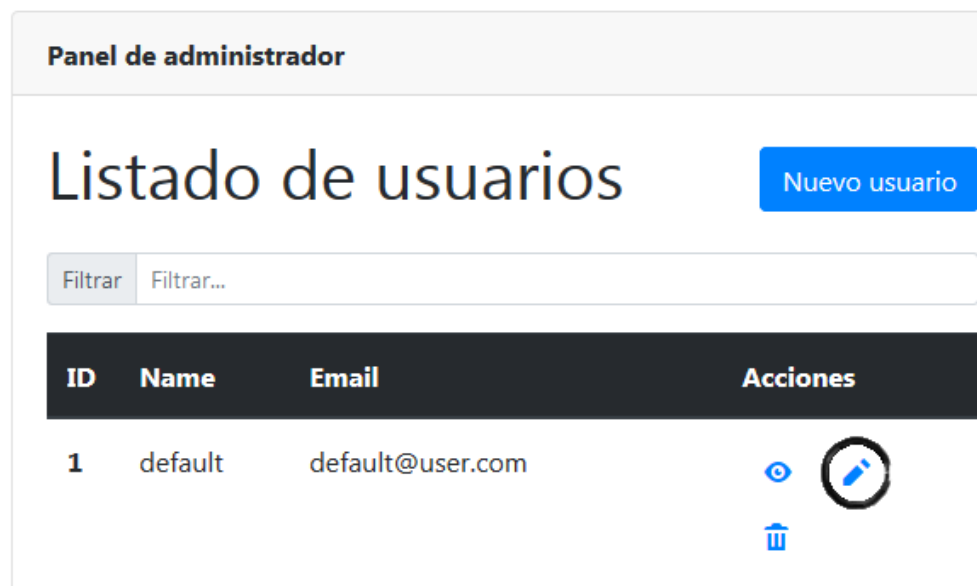
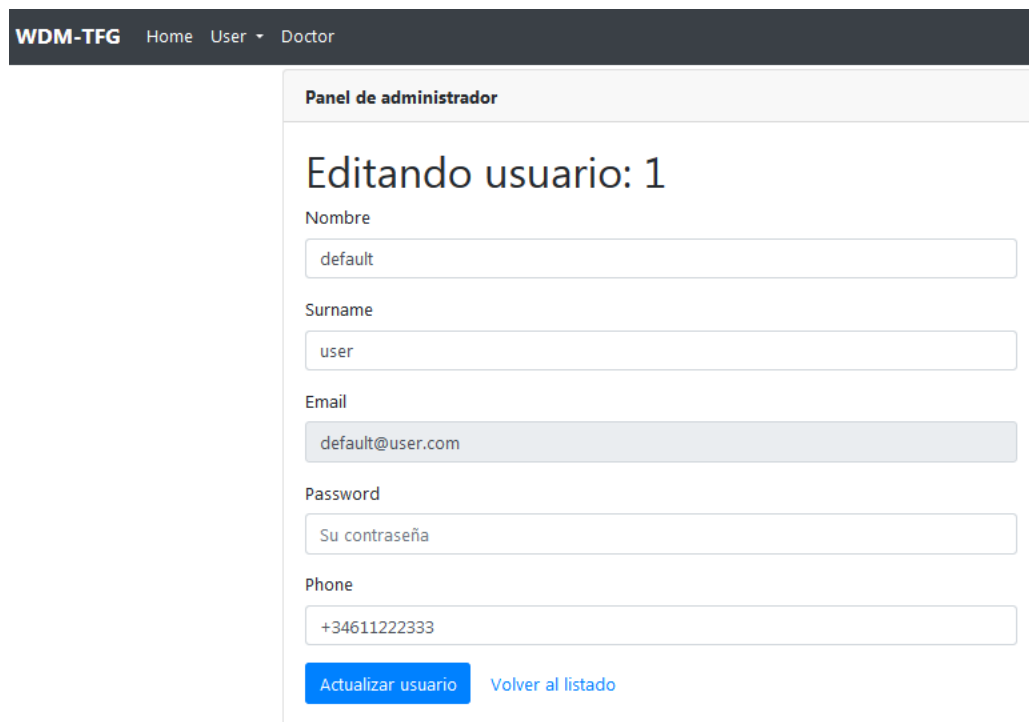


Ilustración 123: Captura del icono de "editar usuario"

En la ilustración 123 es posible observar el botón de editar usuario.





WDM-TFG Home User ▾ Doctor

**Panel de administrador**

### Editando usuario: 1

Nombre

Surname

Email

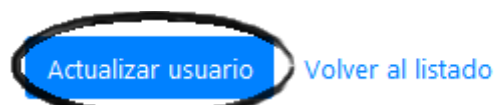
Password

Phone

[Actualizar usuario](#) [Volver al listado](#)

*Ilustración 124: Captura del formulario de edición de un usuario*

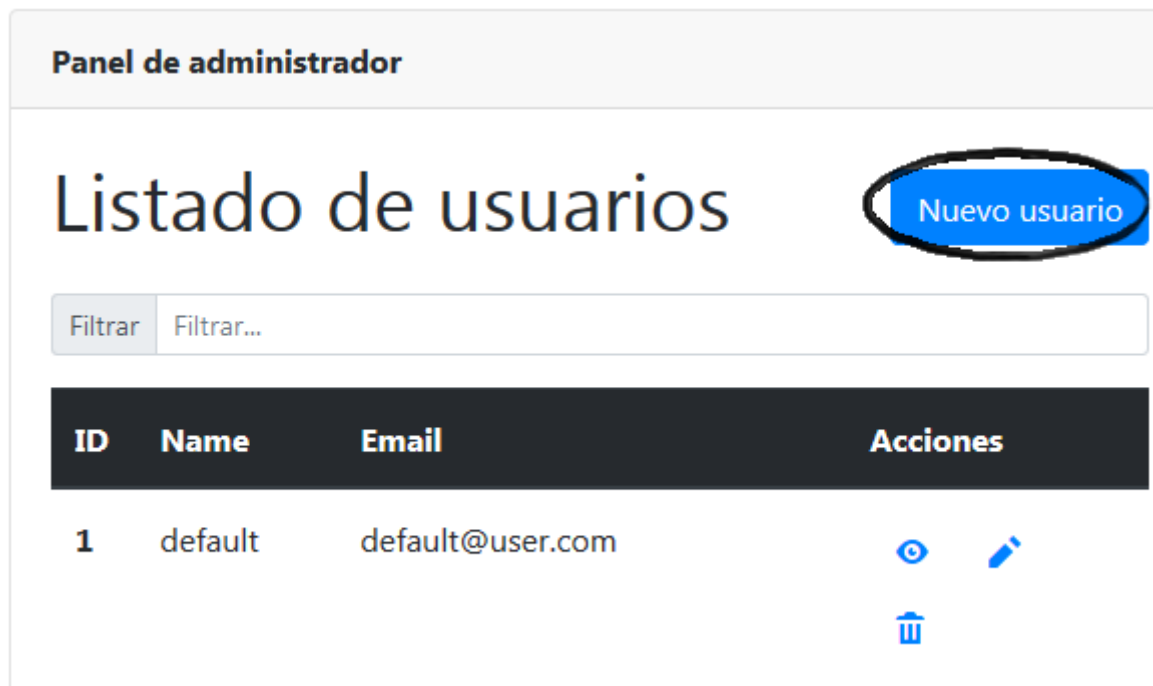
El administrador puede editar todos los campos de un usuario a excepción del email (ilustración 124).



*Ilustración 125: Captura del botón "Actualizar usuario" del formulario de edición de usuarios*

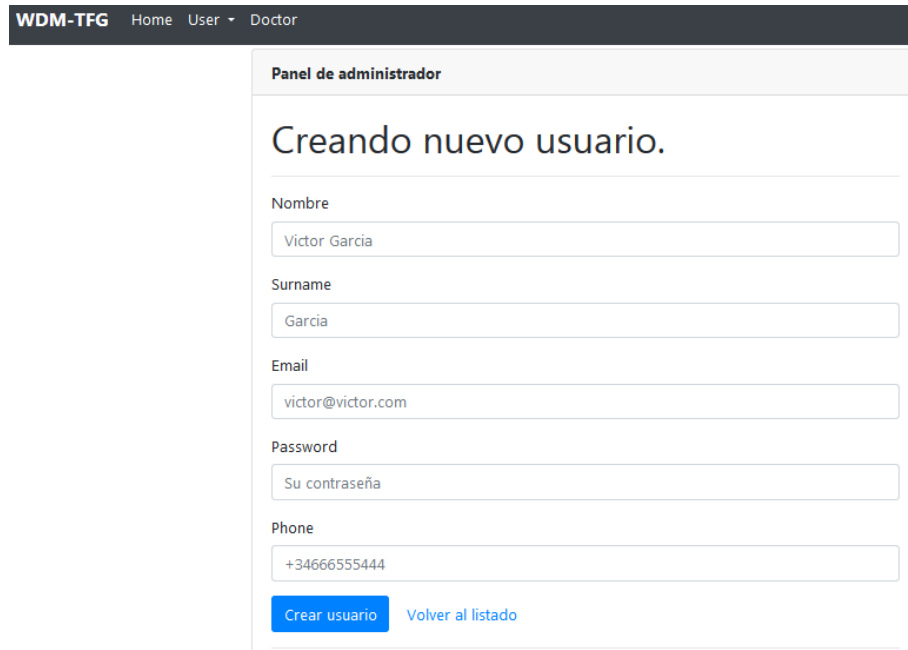
Pulsando el botón “Actualizar usuario” mostrado en la ilustración 125, en el formulario de edición de usuario, se guardarán los cambios realizados.

#### 14.3.4.4 Crear usuario



*Ilustración 126: Captura del botón "Nuevo usuario" del listado de usuarios*

Cuando el administrador se encuentra en el listado de usuarios, puede pulsar el botón “Nuevo usuario” (ilustración 126) para ser dirigido al formulario de creación de usuarios. En dicho formulario podrá rellenar todos los campos correspondientes a la información de usuario y guardarlo en el sistema.



**WDM-TFG** Home User ▾ Doctor

**Panel de administrador**

### Creando nuevo usuario.

Nombre  
Victor Garcia

Surname  
Garcia

Email  
victor@victor.com

Password  
Su contraseña

Phone  
+34666555444

[Crear usuario](#) [Volver al listado](#)

*Ilustración 127: Captura del formulario de creación de usuarios*

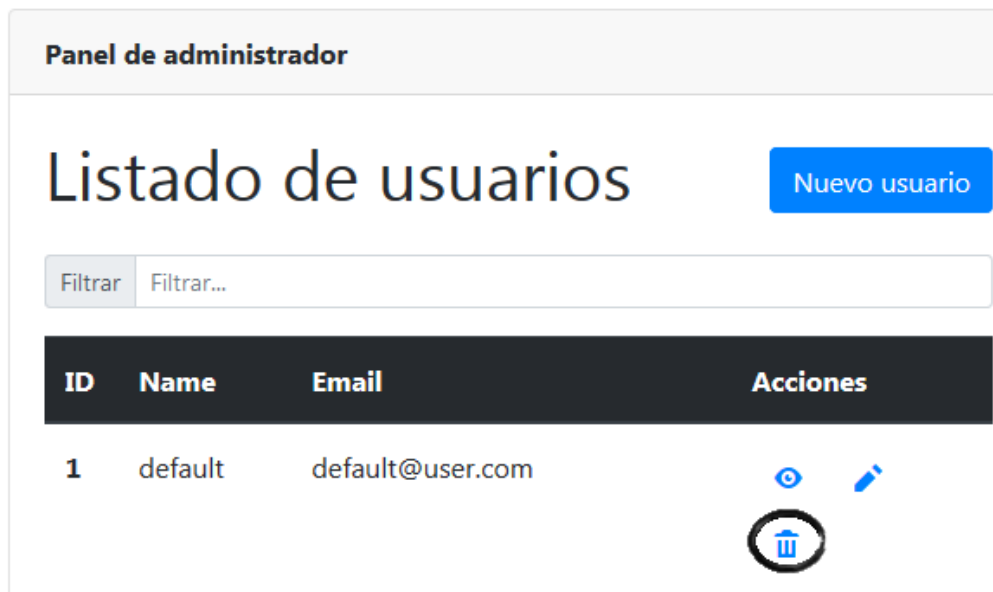
La ilustración 127 muestra el formulario de creación de usuarios.



*Ilustración 128: Captura del botón de "Crear usuario" del formulario de creación de usuarios*

Al pulsar el botón “Crear usuario”, de la ilustración 128, el sistema guardará al nuevo usuario.

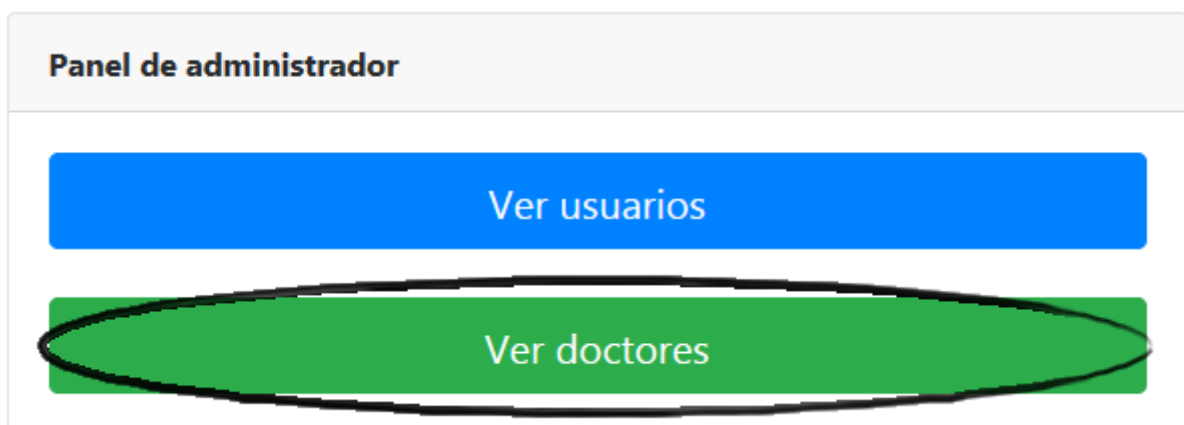
#### 14.3.4.5 Borrar usuario



*Ilustración 129: Captura del icono de "borrar usuario" del listado de usuarios*

Pulsando el icono de “borrar usuario”, señalado en la ilustración 129 en el listado de usuarios, el sistema borrará el usuario indicado.

#### 14.3.4.6 Ver listado de doctores



*Ilustración 130: Captura del botón "Ver doctores" del panel de administrador*

En el panel de administrador, si el administrador pulsa al botón “Ver doctores” indicado en la ilustración 130, se desplazará al listado de doctores.



Ilustración 131: Captura del listado de doctores

El administrador podrá ver una página como la mostrada en la ilustración 131 con el listado de doctores. En esta pantalla aparecen todos los doctores que hay en el sistema y ofrece al admin una serie de acciones a realizar sobre ellos.

A continuación se describen dichas acciones.

#### 14.3.4.7 Ver detalle de doctor



Ilustración 132: Captura del icono "ver detalle" de un doctor

Es posible realizar tres acciones sobre un doctor creado. En este caso, es posible ver el icono "Ver detalles" de un doctor en la ilustración 132.

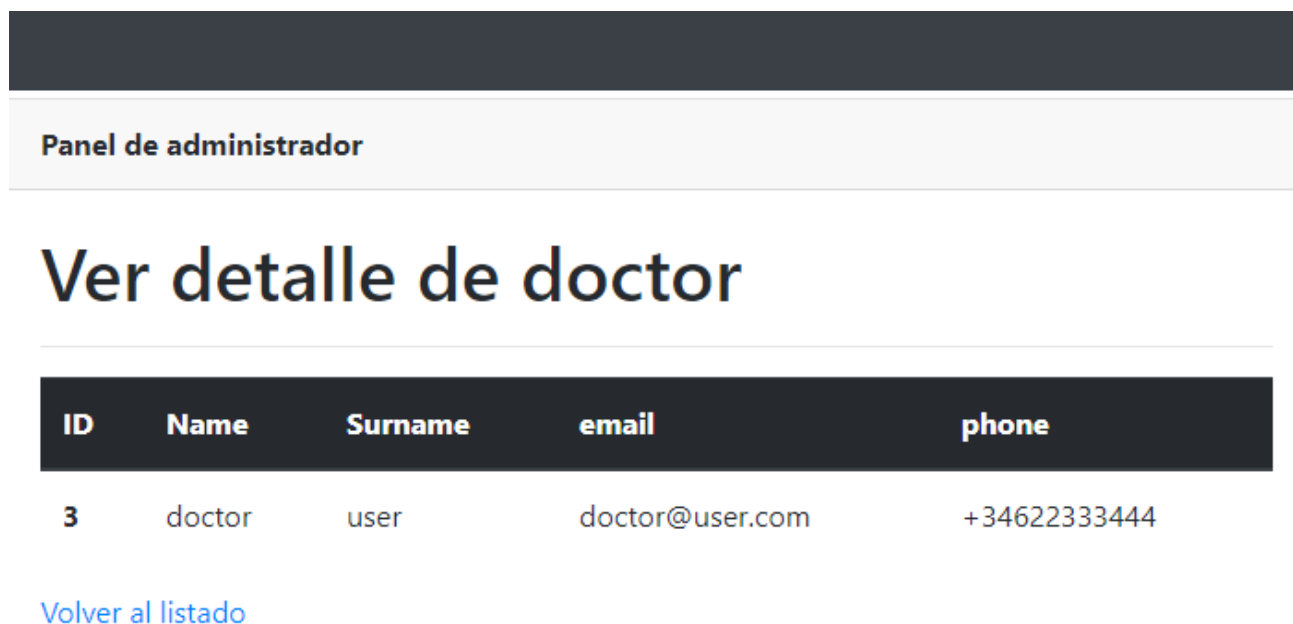


Ilustración 133: Captura de los detalles de un doctor

La página de detalle muestra la información completa de un doctor (ilustración 133).

#### 14.3.4.8 Editar un doctor



Ilustración 134: Captura del icono de "editar doctor"

En la ilustración 134 es posible observar el botón de editar doctor.

**WDM-TFG** Home User ▾ Doctor

**Panel de administrador**

### Editando doctor: 2

Nombre

Surname

Email

Password

Phone

[Actualizar doctor](#) [Volver al listado](#)

Ilustración 135: Captura del formulario de edición de un doctor

El administrador puede editar todos los campos de un doctor a excepción del email (ilustración 135).



*Ilustración 136: Captura del botón "Actualizar doctor" del formulario de edición de doctores*

Pulsando el botón “Actualizar doctor” mostrado en la ilustración 136, en el formulario de edición de doctor, se guardarán los cambios realizados.

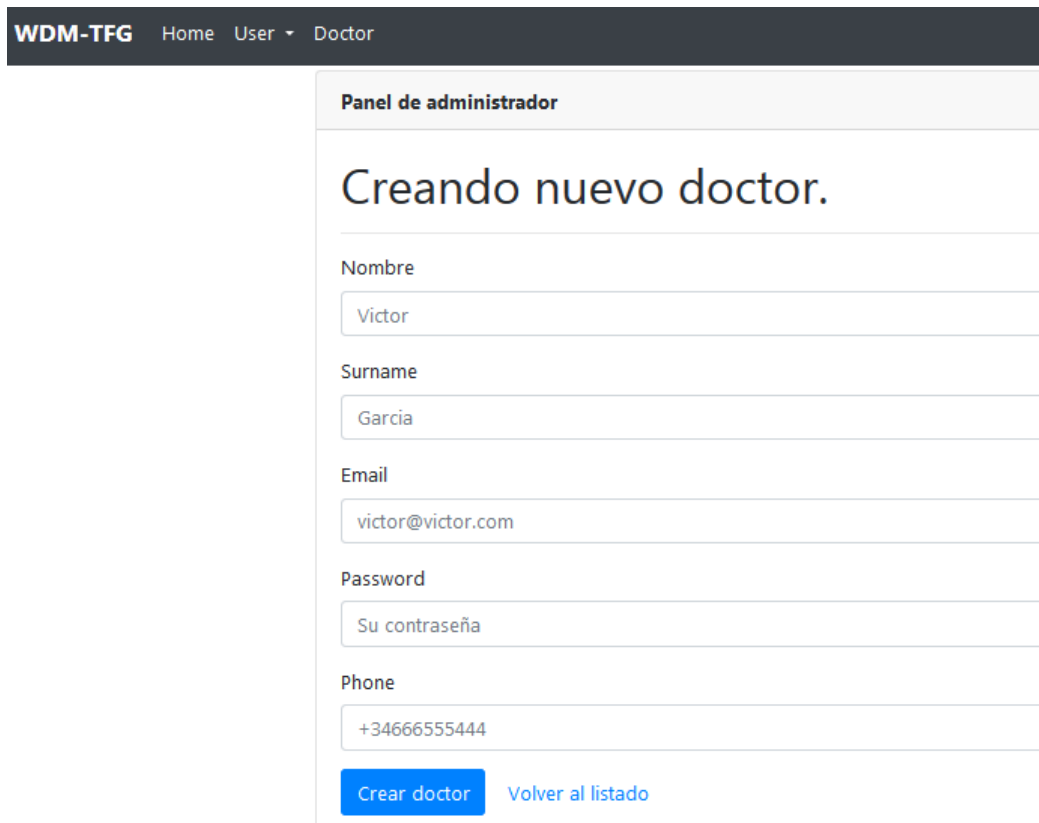
#### 14.3.4.9 Crear usuario



*Ilustración 137: Captura del botón "Nuevo doctor" del listado de doctores*

Cuando el administrador se encuentra en el listado de doctores, puede pulsar el botón “Nuevo doctor” (ilustración 137) para ser dirigido al formulario de creación de doctores. En dicho formulario podrá rellenar todos los campos correspondientes a la información de doctor y guardarlo en el sistema.





**WDM-TFG** Home User ▾ Doctor

**Panel de administrador**

## Creando nuevo doctor.

Nombre  
Victor

Surname  
Garcia

Email  
victor@victor.com

Password  
Su contraseña

Phone  
+34666555444

[Crear doctor](#) [Volver al listado](#)

*Ilustración 138: Captura del formulario de creación de doctores*

La ilustración 138 muestra el formulario de creación de doctores.



*Ilustración 139: Captura del botón "Crear doctor" del formulario de creación de doctores*

Al pulsar el botón “Crear doctor” (ilustración 139), el sistema guardará al nuevo doctor.

#### 14.3.4.10 Borrar doctor



Ilustración 140: Captura del icono de "borrar doctor" del listado de usuarios

Pulsando el icono de “borrar doctor”, señalado en la ilustración 140 en el listado de doctores, el sistema borrará el doctor indicado.

## 14.4 Muestra de consulta

El objetivo principal del proyecto es generar una plataforma web, la cual aporte una estructura definida para la ejecución de un script Python. La finalidad del script Python, en una implementación futura, es analizar automáticamente cualquier lunar en busca de riesgo de melanomas. Es por este motivo que es muy importante que, bajo la tecnología de framework PHP Laravel que se implementa, funcione perfectamente la ejecución de este script Python.

```
$process = new Process("C:\Users\Victor\Anaconda2\python scripts\image.py ".$filename);
$process->run();
$result = $process->getOutput();
```

Ilustración 141: Captura de la configuración para ejecutar un script Python en un controlador Laravel

Para poder demostrar la correcta configuración (ilustración 141) y funcionamiento de un script Python (teniendo en cuenta que el script de análisis no es competencia de este proyecto), se ha diseñado un pequeño script para leer la imagen que el usuario sube en la consulta y, guardarla realizando una pequeña modificación.

**Panel de usuario**

## Viendo detalle de consulta: 17

Image:




Image Mod:



User ID	Related Query	Zone	Created at	Result	Resolved
1	17	1	2019-01-30 15:00:40	Riesgo de melanoma elevado	No

### Comment

Ilustración 142: Captura con la imagen subida por el usuario (izquierda) y la imagen modificada por el script (derecha)

En la ilustración 142, se puede ver, a la izquierda, la imagen que el usuario ha subido y, a la derecha, la imagen modificada por el script. Este funcionamiento está implementado, tal y como se ha explicado, para que se pueda ver la ejecución del script de prueba. En una futura ampliación, cuando se desarrolle el script de análisis, no hará falta mostrar una imagen modificada, pues el resultado del algoritmo de análisis será el ejemplo de funcionamiento.